

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

\_\_\_\_\_ Віталій РОМАНКЕВИЧ

(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**Дипломний проєкт**

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Спеціалізовані комп'ютерні системи»

зі спеціальності

**123 «Комп'ютерна інженерія»**

на тему: Комп'ютерна система симуляції розповсюдження звукових хвиль в гідросфері

Виконав: студент IV курсу, групи KB-63

Гнатенко Віталій Денисович

\_\_\_\_\_ (підпис)

Керівник доц.каф.СПСКС, с.н.с., к. т. н Боярінова Ю. Є.

\_\_\_\_\_ (підпис)

Консультант нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М

\_\_\_\_\_ (підпис)

Рецензент \_\_\_\_\_

\_\_\_\_\_ (підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Віталій РОМАНКЕВИЧ  
(підпис) (ініціали, прізвище)

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
на дипломний проект студента  
Гнатенка Віталія Денисовича**

1. Тема проекту «Комп'ютерна система симуляції розповсюдження звукових хвиль в гідросфері», керівник проекту Боярінова Юлія Євгенівна, к. т. н., затверджені наказом по університету від «25» травня 2020 р. №1181-С

2. Термін подання студентом проекту \_\_\_\_\_

3. Вихідні дані до проекту: див. Технічне завдання.

4. Зміст пояснювальної записки

- аналіз існуючих рішень систем симуляції;
- моделювання розповсюдження звукових хвиль;
- розробка компонентів комп'ютерної системи;
- курівництво користувача комп'ютерної системи;
- результати роботи комп'ютерної системи.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

- Взаємозв'язок між модулями програми. Структурна схема.
- Метод розрахунку трасування звукових променів. Схема алгоритму.
- Пошук наявності координат в зоні освітленості. Схема алгоритму.
- Робота модулю графічного зображення результатів. Схема алгоритму.

#### 6. Консультанти розділів проекту\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормконтроль	Клятченко Я.М., доцент		

#### 7. Дата видачі завдання 1.11.2019

##### Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення літератури за тематикою проекту	02.11.2019	
2.	Розроблення та узгодження технічного завдання	20.02.2020	
3.	Аналіз існуючих рішень	10.03.2020	
4.	Підготовка матеріалів першого розділу дипломного проекту	17.04.2020	
5.	Підготовка матеріалів другого розділу дипломного проекту	26.04.2020	
6.	Підготовка матеріалів третього розділу дипломного проекту	04.05.2020	
7.	Підготовка матеріалів четвертого розділу дипломного проекту	12.05.2020	
8.	Підготовка матеріалів п'ятого розділу дипломного проекту	14.05.2020	
9.	Підготовка графічної частини дипломного проекту	18.05.2020	
10.	Оформлення документації дипломного проекту	20.05.2020	
11.	Попередній огляд матеріалів диплому на кафедрі	25.05.2020	

Студент

\_\_\_\_\_

(підпис)

Віталій ГНАТЕНКО

Керівник проекту

\_\_\_\_\_

(підпис)

Юлія БОЯРІНОВА

---

\* Консультантом не може бути зазначено керівника дипломного проекту.

## АНОТАЦІЯ

Дипломний проект включає пояснювальну записку (64 с., 29 рис., 4 додатки).

Об'єкт розробки – комп'ютерна система симуляції розповсюдження звукових хвиль в гідросфері.

Розроблена комп'ютерна система дозволяє:

- моделювати розповсюдження звукових хвиль в водному середовищі;
- виконувати обчислення для семи типів вертикального розповсюдження швидкості звуку;
- виконувати обчислення для будь-якої глибини випромінення;
- розраховувати траєкторію променів після відбиття від поверхонь;
- працювати, як незалежна комп'ютерна система;
- працювати у складі спеціального автоматизованого комплексу.

В комп'ютерній системі передбачена робота з буфером, в якому знаходяться попередньо розраховані данні, що поліпшує швидкодію всієї системи в цілому. В процесі розробки була використана мова програмування C++ разом з фреймворком Qt та бібліотекою побудови графіків QCustomPlot.

В ході виконання дипломного проекту:

- проведено аналіз існуючих рішень;
- розроблено архітектуру системи симуляції розповсюдження звукових хвиль в гідросфері;
- перевірено роботу, як в незалежному, так і інтегрованому в спеціальний автоматизований комплекс режимі.

Приведено необхідні схеми, документація та підведено підсумки щодо виконаної роботи.

**Ключові слова:** комп'ютерна система, інтерфейс користувача, C++, Qt, QML, QCustomPlot.

## **ABSTRACT**

The diploma project includes an explanatory note (64 p., 29 fig., 4 applications).

The object of development is a computer system for simulating the propagation of sound waves in the hydrosphere.

The developed computer system allows:

- model the propagation of sound waves in the aquatic environment;
- perform calculations for seven types of vertical speed propagation of sound;
- perform calculations for any depth of radiation;
- calculate the trajectory of rays after reflection from surfaces;
- work as an independent computer system;
- work as part of a special automated complex.

A computer system provides a buffer that contains pre-calculated data, which improves the performance of the whole system. The C++ programming language was used in the development process along with the Qt framework and the QCustomPlot graphics library.

During the development of the diploma project:

- analysis of existing solutions;
- the architecture of the system of simulation of sound wave propagation in the hydrosphere is developed;
- tested operation, both in independent and integrated in a special automated complex mode.

The necessary schemes, documentation and results on the performed work are given.

Keywords: computer system, user interface, C ++, Qt, QML, QCustomPlot.

[illegible]

					ІАЛЦ. 045490.001 ОА							
Змін.	Арк.	№ докум.	Підпис	Дата	Комп'ютерна система симуляції розповсюдження звукових хвиль в гідросфері  Опис альбому				Літ.	Аркуш	Аркушів	
Розробив	Гнатенко В.Д.										1	2
Перевірів	Боярінова Ю.Є.											
Н. контроль	Клятченко Я.М.											
Затвердив	Романкевич В.О.								КПІ ім. Ігоря Сікорського, ФПМ, КВ-63			

[illegible]

## ЗМІСТ

1.	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2.	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3.	МЕТА І ПРИЗНАЧЕННЯ РОБОТИ .....	2
4.	ДЖЕРЕЛА РОЗРОБКИ.....	2
5.	ТЕХНІЧНІ ВИМОГИ.....	2
5.1	Вимоги до програмного продукту, що розробляється.....	2
5.2	Вимоги до апаратного забезпечення.....	3
5.3	Вимоги до програмного та апаратного забезпечення користувача.....	3
6.	ЕТАПИ РОЗРОБКИ .....	4

					<b>ІАЛЦ.045480.002 ТЗ</b>			
Зм.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Гнатенко В.Д.			Комп'ютерна система симуляції розповсюдження звукових хвиль в гідросфері <b>Пояснювальна записка</b>			
Перевір.		Боярінова Ю.Є						
Н. контр.		Клятченко Я.М.			Літ.    Аркуш    Аркушів 1                    4 <b>КП ім. Ігоря Сікорського ФПМ, КВ-63</b>			
Затв.		Романкевич В.О.						



## 1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Комп'ютерна система симуляції розповсюдження звукових хвиль в гідросфері».

Галузь застосування: інше програмне забезпечення.

## 2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проєктування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

## 3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення комп'ютерної системи симуляції розповсюдження звукових хвиль в гідросфері.

## 4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1 Вимоги до програмного продукту, що розробляється

- сумісність з операційними системами Windows, Linux та Mac;
- можливість роботи системи як незалежно, так і у складі спеціального автоматизованого комплексу для навчального класу підготовки операторів гідроакустичних засобів;
- динамічне виведення результатів розрахунків;
- висока швидкодія;
- можливість взаємодії користувача за допомогою зручного графічного інтерфейсу.

					ІАЛЦ. 045480.002 ПЗ	Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

## **5.2 Вимоги до апаратного забезпечення**

- Процесор: Intel Celeron N4000;
- Оперативна пам'ять: 4 Гб;

## **5.3 Вимоги до програмного та апаратного забезпечення користувача**

- Операційна система Windows, Linux чи Mac;

					ІАЛЦ. 045480.002 ПЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

## 6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	02.11.2019
2.	Розроблення та узгодження технічного завдання	20.02.2020
3.	Аналіз існуючих рішень	10.03.2020
4.	Підготовка матеріалів першого розділу дипломного проекту	17.04.2020
5.	Підготовка матеріалів другого розділу дипломного проекту	26.04.2020
6.	Підготовка матеріалів третього розділу дипломного проекту	04.05.2020
7.	Підготовка матеріалів четвертого розділу дипломного проекту	12.05.2020
8.	Підготовка матеріалів п'ятого розділу дипломного проекту	14.05.2020
9.	Підготовка графічної частини дипломного проекту	18.05.2020
10.	Передзахист дипломного проекту	20.05.2020

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 045480.002 ПЗ

Арк.

4

# ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ .....	4
ВСТУП .....	5
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ СИСТЕМ СИМУЛЯЦІЇ РОЗПОВСЮДЖЕННЯ ЗВУКОВИХ ХВИЛЬ .....	6
1.1 Основні поняття предметної області.....	6
1.2 Аналіз існуючих рішень.....	12
1.3 Обґрунтування теми дипломного проекту.....	15
Висновки до розділу.....	16
2. МОДЕЛЮВАННЯ РОЗПОВСЮДЖЕННЯ ЗВУКОВИХ ХВИЛЬ .....	17
2.1 Рівняння акустичної хвилі.....	17
2.2 Формалізм Лагранжа та принцип Ферма.....	19
2.3 Сповільнення швидкості звукових променів.....	20
2.4 Циліндрична симетрія. Двовимірний випадок.....	21
2.5 Апроксимація променів Гауса.....	23
2.6 Загасання.....	26
2.7 Чисельні методи.....	28
2.7.1 Метод Рунге-Кутта-Фелберга .....	28
2.7.2 Розрахунок рівнянь Ейконала.....	29
2.7.3 Розрахунок точок після відбиття.....	30
2.7.4 Корекція за рахунок явища рефракції.....	32

					ІАЛЦ. 467200.002 ПЗ			
Змін	Арк.	№ докум.	Підпис	Дата				
Розробив		Гнатенко В.Д.			Комп'ютерна система симуляції розповсюдження звукових хвиль в гідросфері  <b>Пояснювальна записка</b>	Літ.	Аркуш	Аркушів
Перевірів		Боярінова Ю.Є.					1	64
						КПІ ім. Ігоря Сікорського, ФПМ КВ-63		
Н. контроль								
Затвердив								

Висновки до розділу.....	33
3. РОЗРОБКА КОМПОНЕНТІВ КОМП'ЮТЕРНОЇ СИСТЕМИ.....	34
3.1 Особливості реалізації та використання додаткових модулів.....	34
3.2 Головний модуль.....	38
3.2.1 Підключення сторонньої бібліотеки побудови графіків.....	39
3.2.2 Підключення головного модулю .....	41
3.3 Модуль симуляції розповсюдження звукових хвиль.....	42
3.3.1 Модуль розрахунків симуляції .....	42
3.3.2 Модуль звукового профілю.....	45
3.3.3 Модуль задання значень за замовчуванням .....	46
3.3.4 Модуль рельєфу .....	47
3.3.5 Модуль основних даних .....	48
3.3.6 Модуль розрахунків траєкторії розповсюдження променів.....	48
3.4 Модуль графічного зображення результатів симуляції.....	50
3.5 Модуль графічного інтерфейсу користувача.....	52
3.6 Загальний алгоритм роботи системи.....	52
Висновки до розділу.....	53
4. КЕРІВНИЦТВО ДЛЯ КОРИСТУВАЧА КОМП'ЮТЕРНОЇ СИСТЕМИ.....	54
Висновки до розділу.....	55
5. РЕЗУЛЬТАТИ РОБОТИ КОМП'ЮТЕРНОЇ СИСТЕМИ .....	56
ВИСНОВКИ.....	62

ДОДАТКИ

Додаток 1. Копії графічного матеріалу.

ІАЛЦ.045490.005 Д1. Взаємозв'язок між модулями програми.

Структурна схема.

ІАЛЦ.045490.006 Д2. Метод розрахунку трасування звукових променів.

Схема алгоритму.

ІАЛЦ.045490.007 Д3. Пошук наявності координат в зоні освітленості.

Схема алгоритму.

ІАЛЦ.045490.008 Д4. Робота модулю графічного зображення результатів. Схема алгоритму.

Додаток 2. Фрагменти програмного коду.

Додаток 3. Приклад застосування системи (стаття).

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ВРШС – вертикальний розподіл швидкості звуку

Гідроакустика – це розділ акустики, що вивчає випромінення, розповсюдження та прийом звукових хвиль в водному середовищі

ЗТ – зона тіні

ПЗ – програмне забезпечення

ПЗК – підводний звуковий канал

Пеленг — це напрям від спостерігача до цілі, який визначається кутом між площиною меридіана та вертикальною площиною

Рефракція – це явище викривлення шляху розповсюдження звукового променя в зв'язку зміни швидкості поширення звуку в середовищі

					ІАЛЦ.467200.002 ПЗ	Арк.
						4
Змін.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

В умовах сьогодення, людський фактор відіграє величезну роль у забезпеченні успішного функціонування керованих людьми складних технічних систем. Особливо це стосується фахівців, що забезпечують виявлення та класифікацію об'єктів за допомогою інформації, отриманої з моніторів їх робочих місць. Якість роботи сучасних комп'ютерних систем, ефективність їх застосування і, як правило, безпека людей і техніки залежить від професіоналізму та досвіду цих фахівців, їх майстерності і рівня практичних навичок обробки інформації.

Для професійної підготовки операторів у навчальних центрах, університетах та інших освітніх організаціях, що займаються обробкою інформації в складних динамічних системах, використовуються навчально-тренувальні засоби, що призначені для отримання практики ефективної роботи з відтворенням умов, які можуть виникати в їх професійній діяльності.

Зважаючи на триваючу збройну агресію Російської Федерації проти України, тимчасову окупацію території Автономної Республіки Крим, міста Севастополя та окремих районів Донецької і Луганської областей, постала гостра необхідність комплектування Збройних Сил України висококваліфікованим особовим складом. Зокрема, для підготовки у навчальних центрах ВМС України операторів гідроакустичних засобів, було запропоновано розробити тренажер, що забезпечує максимальне наближення процесу навчання операторів гідроакустичних систем надводних кораблів до реальних умов їх роботи на кораблі та дозволить відпрацьовувати ними завдання практичної підготовки.

У дипломному проєкті розглянуто один з основних модулів програмного забезпечення спеціального автоматизованого комплексу для навчального класу підготовки операторів гідроакустичних засобів ВМС України.

					ІАЛЦ.467200.002 ПЗ	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		



# 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ СИСТЕМ СИМУЛЯЦІЇ РОЗПОВСЮДЖЕННЯ ЗВУКОВИХ ХВИЛЬ

## 1.1 Основні поняття предметної області

За останні 100 років експлуатація морів та їхніх ресурсів постійно зростала. Тому виникла необхідність у постійному вдосконаленні дистанційних методів і технологій вивчення водного середовища. Ці методи використовуються з різним ступенем успіху залежно від об'єктів, які слід виявити чи дослідити, та включають:

- магнітометрію ( магнітне виявлення аномалії, де можна виміряти незначні зміни магнітного поля Землі через наявність об'єкта);
- оптичні методи;
- зміни електричного поля;
- гідродинаміки, таких як зміни тиску;
- термічні методи;
- електромагнітні хвилі;
- акустичні хвилі.

Для виявлення цілі на далекій відстані найбільше підходить метод з використанням акустичних хвиль.

Акустичні хвилі виявились дуже корисним інструментом для виявлення ресурсів і предметів у товщі води. В водному середовищі звук розповсюджується значно швидше, ніж в повітрі (близько 1500 м/с) і на більші відстані, внаслідок малого затухання. В середньому, дальність виявлення цілі даним методом в два рази перевищує наступний за дистанцією – магнітометричний метод. Акустичний метод дозволяє виявити підводні човни на всіх глибинах, незалежно від часу доби, погодних умов чи пори року.

На практиці акустичний метод був вперше використаний при побудові гідроакустичної станції «ASDIC» (абревіатура від **Allied Submarine Detection Investigation Committee**) під кінець Першої світової війни в Великій Британії. Принцип її роботи був таким:

					ІАЛЦ.467200.002 ПЗ	Арк.
						6
Змін.	Арк.	№ докум.	Підпис	Дата		

- п'єзовипромінювач під дією струму деформується, створюючи в певному напрямку акустичні хвилі високої частоти;
- акустичний сигнал, зустрівшись з перешкодою, відбивається в зворотньому напрямку і вловлюється гідрофонами;
- заміривши час проходження сигналу, апаратура видає відстань до об'єкту.

Насьогодні, основний принцип дії сучасних сонарів залишився незмінним. Однак, в порівнянні з «ASDIC», теперішні гідролокатори мають поліпшену ефективність, в зв'язку з розширенням масштабів його використання, а також зі збільшенням числа кораблів, з яких сонар міг використовуватися для проведення пошуку й ведення вогню по ворожим підводним човнам.

Найбільш поширеними типами сонарів є пасивний та активний. У пасивному режимі акустичний сигнал зароджується в мішені і поширюється на приймач, де перетворюється на електричний сигнал для обробки й подальших розрахунків. Даний режим ще називають шумопеленгуванням. Позитивною стороною пасивного способу є те, що об'єкт пошуку не знає про той факт, що його помітили, а також доволі швидка кваліфікація цілі за рахунок акустичної сигнатури (шум обладнання, шум гвинтів, шум обтікання, тощо). Але негативною стороною є неможливість отримати інформацію про дистанцію до цілі, можна одержати лише напрямок пеленгу.

В активному режимі електричний сигнал перетворюється передавачем в акустичний сигнал, і звукові хвилі поширюються від передавача до цілі та назад до приймача, де відбувається перетворення з акустичного в електричний сигнал з подальшою обробкою отриманих даних (рис 1.1). Обробка сигналів спрямована на посилення зворотного сигналу від цілі або зменшення шуму, в який може бути вбудований зворотний сигнал. З одного боку, даний спосіб, в порівнянні з шумопеленгуванням, дає змогу безпосередньо дізнатись не тільки напрямок, а й дистанцію до об'єкту, а також швидкість рухомої цілі. Але з

іншого боку, в порівнянні з пасивним методом, дистанція виявлення об'єктів менша, адже відбита звукова хвиля зі збільшенням дистанції затухає і інтенсивності відбитої хвилі не вистачає для коректної роботи апаратури. Також, важливо зазначити, що при генерації звукової хвилі, ворожий підводний човен чує випромінений сигнал на дистанції, що приблизно в два рази більше, ніж гідроакустична станція почує відбитий звуковий сигнал. Незважаючи на ці недоліки, гідроакустика отримала дуже широке практичне застосування, наприклад:

- океанологічні дослідження;
- дресирування морських тварин;
- звуко-підводний зв'язок – система передачі та прийому інформації між підводними човнами та береговими органами управління;
- морська навігація;
- вирішення військових задач, а саме: знаходження підводних чи надводних човнів, торпед, безпілотних підводних апаратів, мін тощо.

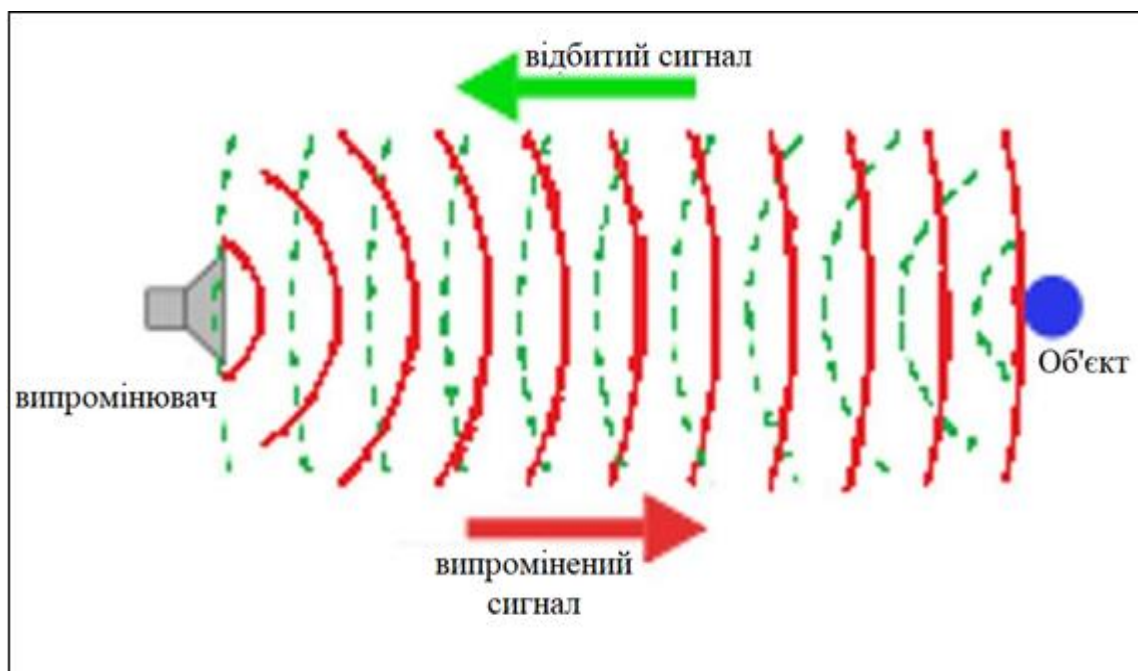


Рисунок 1.1 – Принцип роботи активного типу

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

8

Зазвичай, в гідроакустиці використовують звукові хвилі частотами від 300 до 10000 Гц та ультразвуковими від 10000 Гц і вище. На розповсюдження звукових хвиль впливають багато факторів, таких як:

- солоність;
- пора року;
- температура води;
- глибина водойми.

Ці фактори формують явище рефракції (рис. 1.2). Наприклад: при температурі 30 градусів та солоності 35‰ швидкість звуку дорівнює 1546 м/с. Зі збільшенням солоності на 1‰ швидкість звуку приблизно збільшиться на 1,3 м/с. Але зміна солоності для морського середовища не є головним фактором, що визначає швидкість звуку у морі. У верхніх шарах водного середовища головну роль грає температура та її рівномірність, оскільки при збільшенні температури лише на 1 градус – збільшиться швидкість звуку приблизно на 3 м/с. У глибоких шарах, там де температура постійна, незалежно від пори року, домінує тиск – на кожні 100 м глибини швидкість звуку збільшується орієнтовано на 2 м/с. Якщо б середовище було акустично рівномірним (швидкість звуку була постійною на будь-якій глибині), то явища рефракції не спостерігалось. Звукова хвиля рухалась б в усіх напрямках від джерела з однаковою швидкістю і викривлення фронту не спостерігалось.

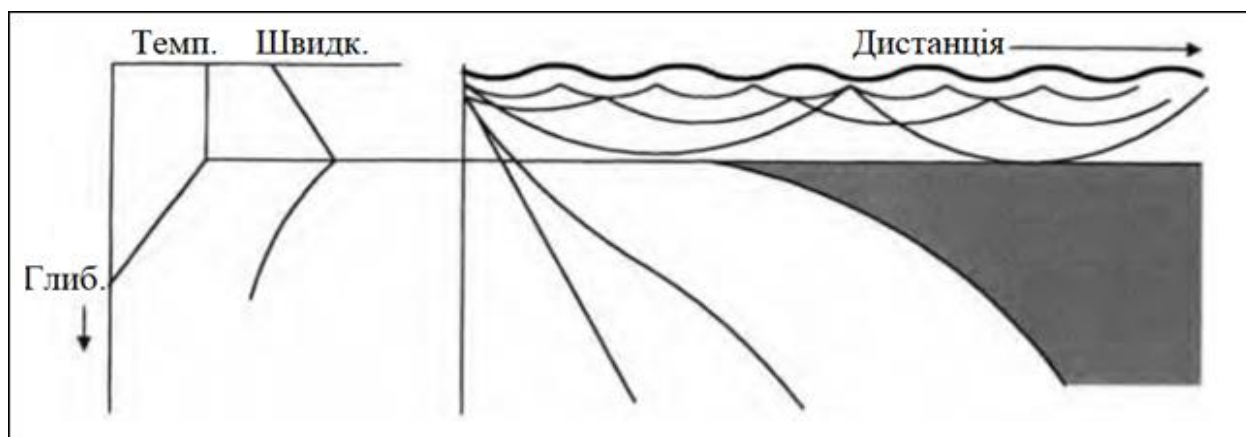


Рисунок 1.2 – Явище рефракції

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

9

Звукові промені завжди повертаються до шару з меншою швидкістю звуку, і чим менша швидкість звуку, тим більше виражається рефракція. В деяких випадках можна натрапити на шар, в якому швидкість звуку найменша, вище цієї глибини швидкість звуку збільшується через нагрів води, а нижче – збільшується внаслідок збільшення тиску, завдяки чому виникає явище підводного звукового каналу.

Підводний звуковий канал – це шар води в морях та океанах, в якому можливе наддалеке поширення звуку внаслідок особливостей рефракції. Глибина, на якій швидкість звуку є мінімальною, називається віссю ПЗК. На рис. 1.3. зображено ПЗК та його вісь пунктирною лінією. Звуковий промінь, що відхилився від осі ПЗК до поверхні води чи до поверхні дна, внаслідок рефракції завжди буде прагнути повернутися назад до осі. Значне збільшення дистанції поширення звуку при наявності ПЗК може спостерігатись і близько до поверхні. В цьому випадку промені, що йдуть донизу під впливом рефракції, заходять в глибоководні шари, де вони відхиляються вгору і виходять знову до поверхні на достатньо великій відстані від джерела. Далі, картина поширення променів повторюється і в результаті утворюється послідовність так званих вторинних освітлених зон. При розповсюдженні в ПЗК звукові хвилі не торкаються ні дна водного середовища, ні поверхні, а отже, не розсіюються і не поглинаються на кордонах середовища.

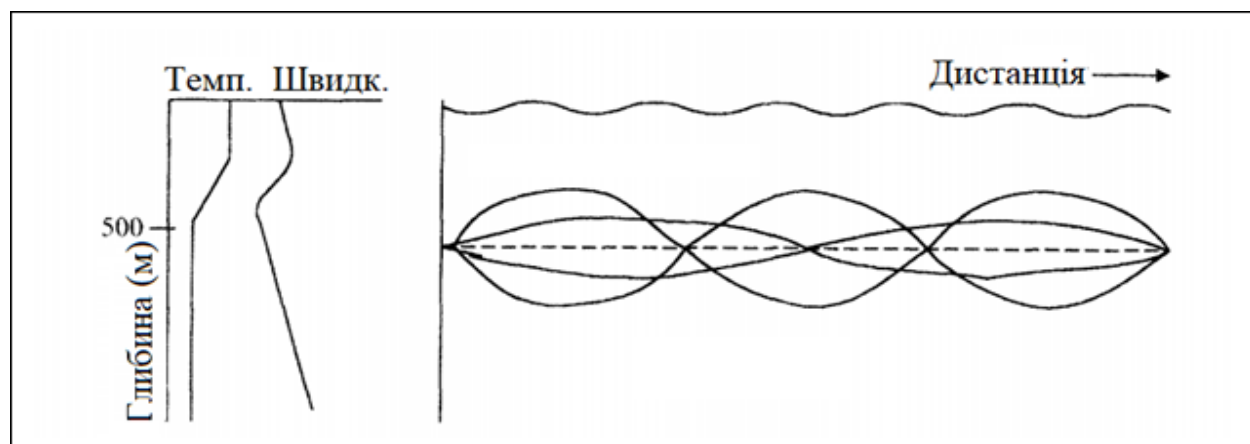


Рисунок 1.3 – Розповсюдження хвилі в ПЗК

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

10

В результаті рефракції на різновіддалених зонах від джерела звуку, можна спостерігати зони акустичної освітленості та тіні (Рис 1.4), які полегшують чи погіршують виявлення підводних об'єктів за допомогою гідроакустичних засобів. Зоною акустичної освітленості є область водного середовища, в якій можливий прийом посилок(сигналів) створених джерелом звуку або відбитих звукових коливань від об'єкту, що спостерігається. У зонах тіні, які утворюються, зазвичай, при негативній рефракції, об'єкт не виявляється або його виявлення значно погіршується в результаті викривлення звукових променів.

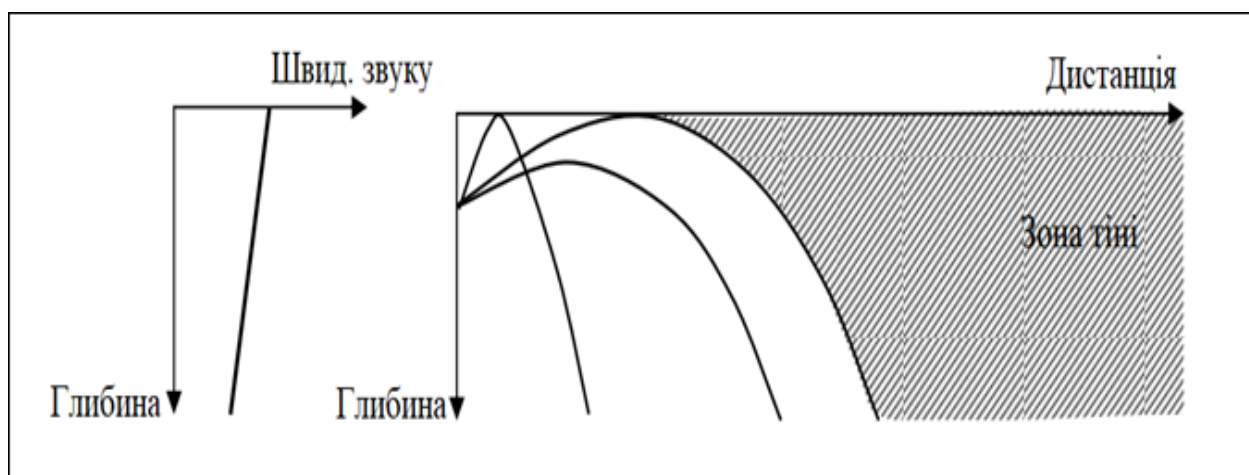


Рисунок 1.4 – Створення зони тіні

На основі досліджень та показників зібраних на багатьох місцевостях було виявлено сім основних типів вертикального розподілу швидкості звуку (ВРШЗ) (рис. 1.5). Кожний тип притаманний тому чи іншому регіону, наприклад: в Чорному морі притаманні перший, третій та шостий типи.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

11

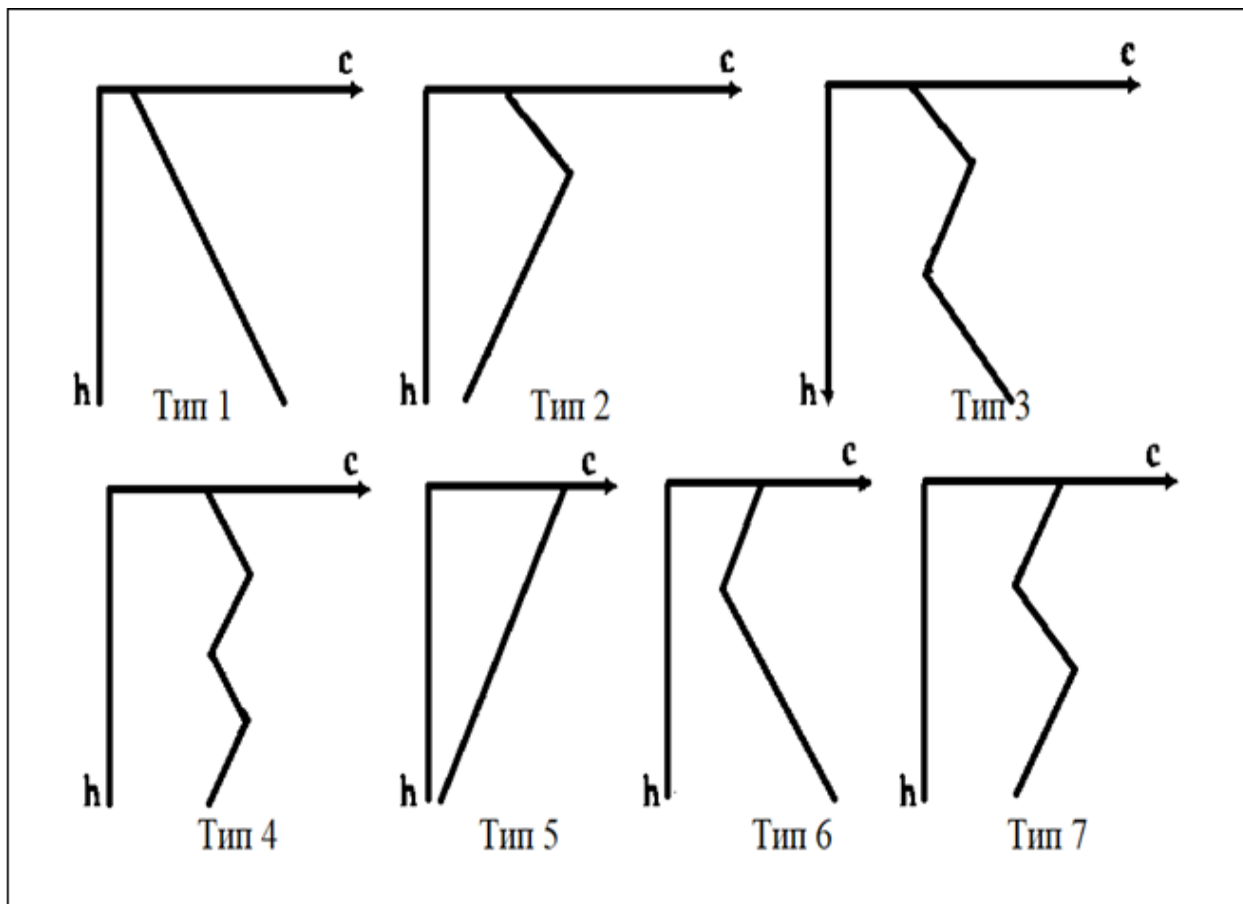


Рисунок 1.5 – Основні типи ВРШЗ

## 1.2 Аналіз існуючих рішень

Під час підготовки до розробки комп'ютерної системи симуляції розповсюдження звукових хвиль, було здійснено аналіз існуючих програмних комплексів та систем. Насьогодні не існує аналогів даній комп'ютерній системі, але є схожі за функціональними можливостями програми.

Нижче наведено приклади деяких відомих програм, що дозволяють віддалено реалізувати функціонал, який виконує дана комп'ютерна система.

**BELLHOP ray tracing program** — програма відстеження променів написана мовою Fortran, як частина акустичного інструментарію. Bellhop була розроблена для того, щоб виконати двовимірну модель відстеження акустичних променів для заданого профілю швидкості звуку  $c(z)$  або задане поле швидкості звуку  $c(r, z)$ , в океанських хвильоводах з плоскими або змінними поглинаючими межами. Параметри виводу включають:

Змін.	Арк.	№ докум.	Підпис	Дата

- координати променю;
- час у дорозі;
- амплітуда;
- власні маси;
- акустичний тиск або втрати передачі (когерентні, некогерентні, напівкогерентні).

Розрахунок акустичного тиску відбувається на основі теорії Гаусових променів [1,2], яку можна застосувати, використовуючи різні наближення, а саме:

- геометричні промені;
- промені з центрованими координатами;
- промені з декартовими координатами;
- гаусова безпроменева апроксимація.

Також BELLHOP реалізована у Fortran, Matlab, Python та використовується на декількох платформах (Mac, Windows та Linux).

**TRACEO** — модель відстеження променів написана мовою Fortran 77, протестована за допомогою компілятора GNU gfortran і доступна за ліцензією Creative Commons. Поточна версія програми TRACEO може обробляти певний набір аналітичних профілів швидкості звуку або загальні табличні профілі швидкості звуку, або поля (рис 1.6). Ресивер приймача може бути горизонтальним, вертикальним, прямокутним, або він може мати довільну криволінійну форму; гідрофони не повинні мати однакову відстань. Промені можуть бути частково або повністю відображені на будь-якій межі хвилеводу або бути повністю поглиненими. TRACEO була розроблена для моделювання акустичного розповсюдження в середовищах, з якими інші доступні моделі не змогли впоратися (хвилясті поверхні, складна батиметрія, зміни глибини та діапазону швидкості звуку тощо), а також для застосування в областях геоакустики, вектор сенсорні масиви, підводні комунікації та акустичні бар'єри. TRACEO скористалася напрацюваннями, що наявні у моделі



трасування променів Bellhop, однієї із складових акустичної панелі інструментів. TRACEO запозичує багато методів у Bellhop, але виходить за рамки його можливостей, дозволяючи проводити розрахунки в таких випадках:

- використання набору аналітичних профілів;
- позиціонування цілей між джерелом та масивом приймачів;
- врахування меж із властивостями, залежними від діапазону (які також враховують швидкість зсуву та ослаблення);
- врахування меж, які крім того, можуть частково або повністю відбивати, а можуть повністю поглинути звуковий промінь.

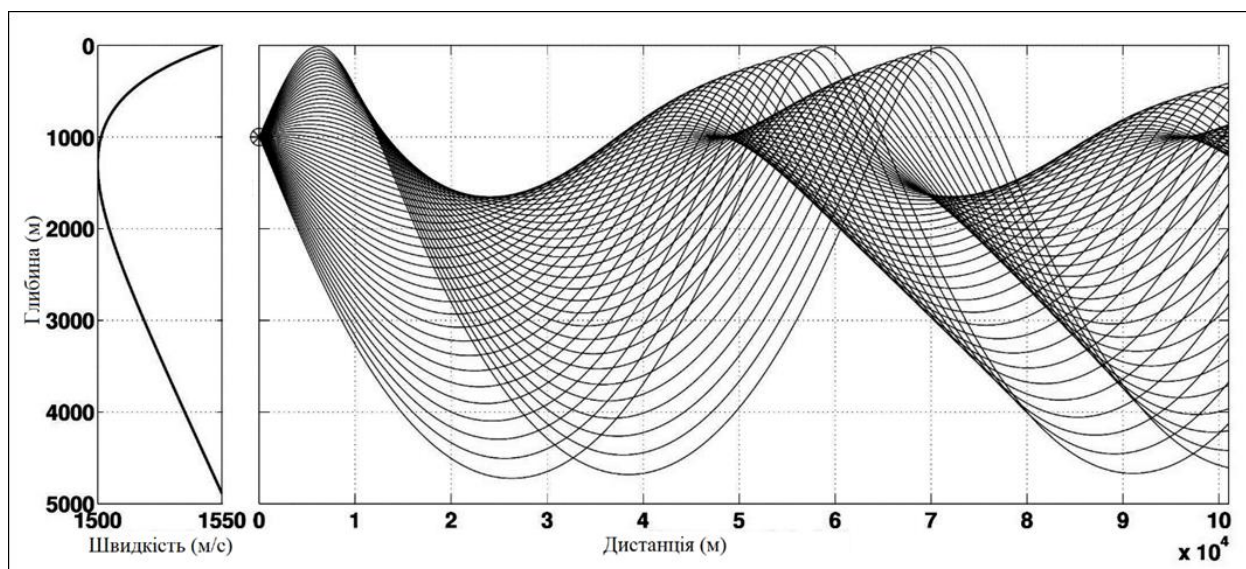


Рисунок 1.6 - Графік на основі вихідних даних TRACEO

Усі вище зазначені програми націлені виключно на дослідницьку діяльність і їх використання не є доцільним, адже вони не можуть забезпечити швидкої роботи і взаємодія з ними потребує спеціальних навичок. Виходячи з цього, можна зробити висновок, що задача розробки комп'ютерної системи симуляції розповсюдження звукових хвиль в гідросфері з високою швидкодією та інтуїтивно зрозумілим інтерфейсом взаємодії є актуальною.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

14

### 1.3 Обґрунтування теми дипломного проекту

На даний час майбутні спеціалісти з гідроакустичних систем у навчальних центрах ВМФ України потребують нового обладнання з програмним забезпеченням для формувань практичних умінь та навичок. В результаті впровадження спеціалізованого автоматизованого комплексу для навчального класу підготовки операторів гідроакустичних систем (рис 1.7.) забезпечується максимально наближене до реальних умов на кораблі навчання. Даний комплекс дозволяє відпрацювати практичні навички особового складу ВМФ з пошуку та ідентифікації підводних та надводних об'єктів, незалежно від їх типу, швидкості чи глибини. Також є можливість в режимі гідролокації та шумопеленгування з корегуванням ПЗ в умовах змін технічних характеристик цілі. Навчальний комплекс дозволяє здійснювати пошук та супроводження цілей в автоматичному та ручному режимах при використанні ціллю засобів гідроакустичної протидії, визначати дистанцію та пеленг цілі.

Для реалізації імітації реальних гідроакустичних умов з'явилась потреба спроектувати та розробити комп'ютерну систему, яка б дозволила імітувати близьку до реальної картини поведінку звукових хвиль в неоднорідному водному середовищі.



Рисунок 1.7 – Зображення автоматизованого комплексу

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

15

Висновки до розділу

В даному розділі розглянуті теоретичні відомості та порівняння з аналогами. Виходячи з того, що аналогів на території України не існує, актуальність створення такої комп'ютерної системи є доцільною.

В якості мови програмування для реалізації даної системи була обрана мова C++. При розробці комп'ютерної системи використовувалась не лише стандартна бібліотека мови програмування C++, а й додатковий фреймворк – Qt зі сторонньою бібліотекою будування графіків. Даний фреймворк є кросплатформовим, що дозволяє запускати розроблені з допомогою ПЗ в більшості сучасних операційних систем. Містить в собі усі основні інструменти для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу та закінчуючи класами роботи з базами даних та мережею [3].

					ІАЛЦ.467200.002 ПЗ	Арк.
						16
Змін.	Арк.	№ докум.	Підпис	Дата		

## 2. МОДЕЛЮВАННЯ РОЗПОВСЮДЖЕННЯ ЗВУКОВИХ ХВИЛЬ

У цій темі буде розглянуто огляд теорії розповсюдження звукових променів в морському середовищі, на якій базуються розрахунки даної комп'ютерної системи.

### 2.1 Рівняння акустичної хвилі

Для початку, перед тим як заглиблюватись у тему моделювання розповсюдження звукових хвиль у водному середовищі, потрібно дізнатись, що представляє собою хвиля в математичному представленні. Для реалізації даного рівняння були використані дослідження, які представлені в бібліотеці TRACEO. Хвильове рівняння має такий вигляд [4]:

$$\nabla^2 p - \frac{1}{c^2} \times \frac{\partial^2 p}{\partial t^2} = S(r_0, \omega), \quad (2.1)$$

де  $p(r, t)$  – тиск акустичної хвилі;

$s(r_0, \omega)$  – сигнал, що передається джерелом акустичного випромінення;

$r_0$  – вказує положення джерела;

$\nabla$  – диференціальний оператор Гамільтона (оператор набла).

За допомогою перетворення Фур'є перетворюємо обидві сторони рівняння (2.1) на рівняння Гельмгольца:

$$\left[ \nabla^2 + \frac{\omega^2}{c^2} \right] P(r, \omega) = S(r_0, \omega), \quad (2.2)$$

де

$$P(r, \omega) = \int_{-\infty}^{\infty} p(r, t) \times e^{-i\omega t} dt,$$

та

$$S(r_0, \omega) = \int_{-\infty}^{\infty} s(r, t) \times e^{-i\omega t} dt$$

Розглянемо одне з наближених рішень рівняння 2.2, що було запропоноване в [5]:

$$P(\omega) = A \times e^{-i\omega\tau}, \quad (2.3)$$

де  $A$  – амплітуда хвилі, що повільно зменшується;

$\omega\tau$  – швидкість розвитку фази.

Поверхні з постійним значенням  $\omega\tau$  називаються хвильовими фронтами, а поверхні з постійним значенням  $\tau$  – часовими фронтами. При об'єднанні рівнянь 2.2 та 2.3 до однорідної форми, беручи до уваги велике наближення частоти, отримуємо:

$$\frac{\nabla^2 A}{A} \ll k^2, \quad (2.4)$$

де  $k = \frac{\omega}{c}$ ;

Під час розділення реального та уявного доданків можна отримати рівняння Ейконала:

$$(\nabla\tau^2) = \frac{1}{c^2}, \quad (2.5)$$

та рівняння транспортування:

$$2 \times (\nabla A \times \nabla\tau) + A\nabla^2\tau = 0. \quad (2.6)$$

В подальшому рівняння Ейконала (2.4) після спрощення має вигляд:

$$\frac{d\tau}{ds} = \frac{1}{c},$$

де  $ds$  означає відстань, пройдену звуковим променем.

З цього можна зробити висновок, що хвиля проходить відстань  $ds$  напрутязі певного часу, а саме поширюється між двома точками А та В і загальний час подорожі тоді дорівнює:

$$\tau = \int_A^B \frac{ds}{c}. \quad (2.7)$$

## 2.2 Формалізм Лагранжа та принцип Ферма

Спираючись на формалізм Лагранжа можна переписати рівняння (2.7) таким чином:

$$\tau = \int_A^B \mathcal{L} ds,$$

де  $\mathcal{L}$  дорівнює  $\frac{1}{c}$ .

У контексті формалізму  $\mathcal{L}$  повинен бути функцією координат і узагальнених швидкостей:

$$\mathcal{L}(x, y, z, \dot{x}, \dot{y}, \dot{z}) = \frac{1}{c} \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}, \quad (2.8)$$

де:

$$\dot{x} = \frac{dx}{ds}, \quad (2.9)$$

$$\dot{y} = \frac{dy}{ds}, \quad (2.10)$$

$$\dot{z} = \frac{dz}{ds}. \quad (2.11)$$

Таким чином, обурення часу подорожі відповідає:

$$\begin{aligned} \delta\tau &= \int_A^B \left\{ \left[ \frac{\partial \mathcal{L}}{\partial x} \delta x + \frac{\partial \mathcal{L}}{\partial \dot{x}} \delta \dot{x} \right] + \left[ \frac{\partial \mathcal{L}}{\partial y} \delta y + \frac{\partial \mathcal{L}}{\partial \dot{y}} \delta \dot{y} \right] + \left[ \frac{\partial \mathcal{L}}{\partial z} \delta z + \frac{\partial \mathcal{L}}{\partial \dot{z}} \delta \dot{z} \right] \right\} ds = \\ &= \left( \frac{\partial \mathcal{L}}{\partial \dot{x}} \delta x + \frac{\partial \mathcal{L}}{\partial \dot{y}} \delta y + \frac{\partial \mathcal{L}}{\partial \dot{z}} \delta z \right) \Big|_A^B + \end{aligned}$$

$$+ \int_A^B \left\{ \left[ \frac{\partial \mathcal{L}}{\partial x} - \frac{d}{ds} \cdot \frac{\partial \mathcal{L}}{\partial \dot{x}} \right] \delta x + \left[ \frac{\partial \mathcal{L}}{\partial y} - \frac{d}{ds} \cdot \frac{\partial \mathcal{L}}{\partial \dot{y}} \right] \delta y + \left[ \frac{\partial \mathcal{L}}{\partial z} - \frac{d}{ds} \cdot \frac{\partial \mathcal{L}}{\partial \dot{z}} \right] \delta z \right\} ds. \quad (2.12)$$

Спираючись на принцип Ферма, а саме на те, що  $\delta \tau = 0$ , та, беручи до уваги, що перша частина рівняння 2.12 дорівнює нулю, отримуємо систему рівнянь:

$$\begin{cases} \frac{d}{ds} \left( \frac{\partial \mathcal{L}}{\partial \dot{x}} \right) - \frac{\partial \mathcal{L}}{\partial x} = 0, \\ \frac{d}{ds} \left( \frac{\partial \mathcal{L}}{\partial \dot{y}} \right) - \frac{\partial \mathcal{L}}{\partial y} = 0, \\ \frac{d}{ds} \left( \frac{\partial \mathcal{L}}{\partial \dot{z}} \right) - \frac{\partial \mathcal{L}}{\partial z} = 0. \end{cases} \quad (2.13)$$

З іншого боку, за допомогою рівняння можна отримати дану систему:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \dot{x}} = \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}} \cdot \mathcal{L} = \mathcal{L} \dot{x} = \frac{1}{c} \frac{dx}{ds}, \\ \frac{\partial \mathcal{L}}{\partial \dot{y}} = \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}} \cdot \mathcal{L} = \mathcal{L} \dot{y} = \frac{1}{c} \frac{dy}{ds}, \\ \frac{\partial \mathcal{L}}{\partial \dot{z}} = \frac{\dot{z}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}} \cdot \mathcal{L} = \mathcal{L} \dot{z} = \frac{1}{c} \frac{dz}{ds}. \end{cases} \quad (2.14)$$

Тому можна зробити висновок, що рішення рівняння Ейконала вимагає рішення наступної системи рівнянь:

$$\begin{cases} \frac{d}{ds} \left( \frac{1}{c} \cdot \frac{dx}{ds} \right) = \frac{\partial}{\partial x} \left( \frac{1}{c} \right), \\ \frac{d}{ds} \left( \frac{1}{c} \cdot \frac{dy}{ds} \right) = \frac{\partial}{\partial y} \left( \frac{1}{c} \right), \\ \frac{d}{ds} \left( \frac{1}{c} \cdot \frac{dz}{ds} \right) = \frac{\partial}{\partial z} \left( \frac{1}{c} \right). \end{cases} \quad (2.15)$$

### 2.3 Сповільнення швидкості звукових променів

Зворотна швидкість звуку систематично присутня в заданій системі за рівнянням (2.15). Такий факт говорить про те, що систему можна значно спростити за допомогою визначення зворотної швидкості звуку як власного параметра, що називається sound slowness (звукова повільність)  $\sigma$ :

$$\sigma = \frac{1}{c}. \quad (2.16)$$

З даним параметром система рівнянь 2.15 має вигляд:

$$\begin{cases} \frac{d}{ds} \left( \sigma \frac{dx}{ds} \right) = \frac{\partial \sigma}{\partial x}, \\ \frac{d}{ds} \left( \sigma \frac{dy}{ds} \right) = \frac{\partial \sigma}{\partial y}, \\ \frac{d}{ds} \left( \sigma \frac{dz}{ds} \right) = \frac{\partial \sigma}{\partial z}. \end{cases} \quad (2.17)$$

Додаткового спрощення можна досягти, розглядаючи звукову повільність звуку як векторний параметр:

$$\sigma = [\sigma_x, \sigma_y, \sigma_z]. \quad (2.18)$$

Спираючись на (2.18), кожний вираз в середині дужок має наступний вигляд:

$$\frac{dx}{ds} = \frac{\sigma_x}{\sigma}, \quad \frac{dy}{ds} = \frac{\sigma_y}{\sigma}, \quad \frac{dz}{ds} = \frac{\sigma_z}{\sigma}, \quad (2.19)$$

що трансформує систему (2.17) у дану систему:

$$\begin{cases} \frac{d\sigma_x}{ds} = \frac{\partial \sigma}{\partial x}, \\ \frac{d\sigma_y}{ds} = \frac{\partial \sigma}{\partial y}, \\ \frac{d\sigma_z}{ds} = \frac{\partial \sigma}{\partial z}. \end{cases}, \quad (2.20)$$

та зводиться до більш компактного вигляду у векторній формі:

$$\frac{d\sigma}{ds} = \nabla \sigma. \quad (2.21)$$

#### 2.4 Циліндрична симетрія. Двовимірний випадок

У системі з циліндричною системою, рівняння Ейконала впливає безпосередньо з рівняння 2.20, замінюючи  $x$  на  $r$  та зводячи нанівець будь-яку похідну вздовж осі  $y$ :



$$\begin{cases} \frac{dr}{ds} = \frac{\sigma_r}{\sigma}, \\ \frac{dz}{ds} = \frac{\sigma_z}{\sigma}. \end{cases}, \quad (2.22)$$

і, беручи це до уваги, система рівнянь (2.17) має вигляд:

$$\begin{cases} \frac{d\sigma_r}{ds} = \frac{\partial \sigma}{\partial r}, \\ \frac{d\sigma_z}{ds} = \frac{\partial \sigma}{\partial z}. \end{cases}, \quad (2.23)$$

де  $\sigma_z(s)$  – горизонтальна складова повільності звуку;

$\sigma_z(s)$  – вертикальна складова повільності звуку.

Виходячи з попередніх тверджень, диференціальний час переміщення променю можна записати як:

$$d\tau = \frac{ds}{c} = \sigma ds = \frac{\sigma^2}{\sigma_r} dr, \quad (2.24)$$

і система рівнянь (2.22) разом з системою (2.23) можуть бути переписані:

$$\begin{cases} \frac{dr}{ds} = c(s)\sigma_r(s), \\ \frac{dz}{ds} = c(s)\sigma_z(s). \end{cases}, \quad (2.25)$$

$$\begin{cases} \frac{d\sigma_r}{ds} = -\frac{1}{c^2} \frac{\partial c}{\partial r}, \\ \frac{d\sigma_z}{ds} = -\frac{1}{c^2} \frac{\partial c}{\partial z}. \end{cases} \quad (2.26)$$

або у більш компактній векторній формі:

$$\frac{dr}{ds} = c(s)\sigma(s), \quad \frac{d\sigma}{ds} = -\frac{1}{c^2} \nabla c. \quad (2.27)$$

Беручи до уваги, що швидкість звуку в водному середовищі залежить від глибини, тому горизонтальна складова повільності звуку дорівнює нулю, і це, в комбінації з плоскими границями, дозволяє вивести класичну форму закону Снеліуса (або Снела) вздовж променю:

$$\sigma_r(s) = \frac{\cos \theta(s)}{c(s)} = \text{const.} \quad (2.28)$$

Сукупність початкових умов для вирішення рівняння Ейконала задається таким чином:

$$r(0) = r_0, \quad z(0) = z, \quad \sigma_r(0) = \frac{\cos \theta(0)}{c(0)}, \quad \sigma_z(0) = \frac{\sin \theta(0)}{c(0)}, \quad (2.29)$$

де  $\theta(0)$  – кут запуску;

$[r_0, z_0]$  – вихідне положення;

$c(0)$  – швидкість звуку у вихідному положенні.

## 2.5 Апроксимація променів Гауса

Вхідна точка для апроксимації акустичного поля як променю Гауса задана аналітичним виразом:

$$P(s, n) = \frac{1}{4\pi} \sqrt{\frac{c_0(s) \cos \theta(0)}{c_0(0) \det Q}} \exp \left\{ -i\omega \left[ \tau(s) + \frac{1}{2} (Mn \cdot n) \right] \right\}, \quad (2.30)$$

де  $c_0(s) = c(s, 0)$  – швидкість звуку по траєкторії променю;

$n = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$  – нормаль до променю, представлена у вигляді двовимірного вектору.

Дану нормаль можна представити як:

$$n \cdot e_s = 0,$$

$$\text{де } e_s = \begin{bmatrix} \frac{dx}{ds} \\ \frac{dy}{ds} \\ \frac{dz}{ds} \end{bmatrix} \text{ – унітарний вектор дотичної до променю.}$$

У подальшому проекція повільності звуку вздовж  $n$  буде записана у вигляді  $\sigma_n$ .

Зважаючи на те, що  $n$  – реальний вектор, обидві матриці  $Q$  та  $M$  є комплексними. Тому комплексна частина добутку  $Mn \cdot n$  спонукає Гаусовий розпад амплітуди променю вздовж нормалі, коли реальна частина вносить

фазові корективи у час руху вздовж  $n$ . Крім того, правильний вибір початкових умов для  $Q$  забезпечують те, що  $\det Q \neq 0$  звільнює від апроксимації променів Гауса.

Обидва компоненти  $n$  і  $\sigma_n$  можна вважати такими, що залежать від певного локального набору параметрів променю, а саме: довжини стрілки променів  $s$ , та кутів  $\alpha_1$  та  $\alpha_2$ . У будь-яку точку променя можна ввести набір з трьох ортогональних одиничних векторів, відомих як вектори поляризації. Першим вектором поляризації є  $e_s$ , а інші два поляризаційні вектори, які будуть представлені як  $e_1$  та  $e_2$ , знаходяться в площині, перпендикулярній  $e_s$ , і перпендикулярні один одному. Вектори  $e_1$  та  $e_2$ , визначають можливі орієнтації вектору  $n$  та  $\sigma_n$  в будь-якій координаті  $s$  на промені:

$$n = n_1 e_1 + n_2 e_2, \quad (2.31)$$

$$\sigma_n = \sigma e_1 + \sigma e_2. \quad (2.32)$$

Окрім матриць  $Q$  та  $M$ , апроксимація променів Гауса включає ще дві матриці  $C$  та  $P$ . Усі чотири матриці пов'язані між собою такими відносинами:

$$\frac{d}{ds} Q = C_0 P, \quad \frac{d}{ds} P = -\frac{1}{C_0^2} \cdot C Q, \quad M = \frac{P}{Q}, \quad (2.33)$$

де

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}, \quad (2.34)$$

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}, \quad (2.35)$$

та

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, \quad (2.36)$$

де

$$C_{ij} = \frac{\partial^2 c}{\partial n_i \cdot \partial n_j} . \quad (2.37)$$

Компоненти  $Q$  та  $P$  відповідають частковим похідним нормалі променю та нормалі звукової повільності за допоміжними параметрами  $\alpha_1$  та  $\alpha_2$ :

$$p_{ij} = \frac{\partial \sigma_i}{\partial \alpha_j} , \quad (2.38)$$

$$q_{ij} = \frac{\partial n_i}{\partial \alpha_j} . \quad (2.39)$$

Взагалі кажучи, єдиний спосіб забезпечити, щоб матриці  $Q(s)$  та  $P(s)$  були комплексними, в той час, як  $C(s)$  є реальною матрицею, це шляхом вибору правильних комплексних початкових умов для обох матриць. Як показують результати дослідницької діяльності, такий правильний вибір є предметом гострих дискусій, які, здається, ще не вирішені. Однак існують конкретні варіанти реальних  $Q(0)$  та  $P(0)$ , які є актуальними для обговорення променів Гаусса.

У випадку точкового джерела можуть бути такі умови:

$$P(0) = \begin{bmatrix} \frac{1}{c(0,0,0)} & 0 \\ 0 & \frac{\cos \theta(0)}{c(0,0,0)} \end{bmatrix}, \quad (2.40)$$

$$Q(0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (2.41)$$

де  $c(0,0,0)$  – швидкість звуку початковій точці.

Оскільки розглядається двовимірний випадок, координати  $r$ ,  $z$  та  $\theta$  означають горизонтальну дистанцію, вертикальну дистанцію (глибину) та нахил променю відносно горизонталі відповідно. Вираз променів Гауса в

цьому випадку легко впливає з виразу (2.6), приймаючи за  $n_2 = 0$  та  $n = n_1$ , що в результаті дає вираз:

$$P(s, n) = \frac{1}{4\pi} \sqrt{\frac{c_0(s) \cos \theta(0)}{c_0(0) q_{\perp}(s) q(s)}} \exp \left[ -i\omega \left( \tau(s) + \frac{1}{2} \frac{p(s)}{q(s)} n^2 \right) \right], \quad (2.42)$$

У виразі (2.42)  $p(s)$  та  $q(s)$  взаємопов'язані так званими динамічними рівняннями:

$$\frac{dq}{ds} = c(s)p(s), \quad \frac{dp}{ds} = -\frac{c_{nn}}{c^2} q(s). \quad (2.43)$$

Похідна другого порядку вздовж нормалі може бути записана за допомогою похідних по  $r$  та  $z$  як [6]:

$$C_{nn} = \left( \frac{dr}{dn} \right)^2 C_{rr} + 2 \left( \frac{dr}{dn} \right) \left( \frac{dz}{dn} \right) C_{rz} + \left( \frac{dz}{dn} \right)^2 C_{zz}, \quad (2.45)$$

де

$$C_{rr} = \frac{\partial^2 c}{\partial r^2}, \quad C_{rz} = \frac{\partial^2 c}{\partial r \partial z}, \quad C_{zz} = \frac{\partial^2 c}{\partial z^2}, \quad (2.46)$$

та

$$\frac{dr}{dn} = -\sin \theta, \quad \frac{dz}{dn} = -\cos \theta. \quad (2.47)$$

Ці похідні можна ідентифікувати як компоненти поляризаційного вектора  $e_1$ , які у двовимірному випадку відповідають:

$$e_s(s) = \begin{bmatrix} \cos \theta(s) \\ \sin \theta(s) \end{bmatrix}, \quad (2.48)$$

$$e_1(s) = \begin{bmatrix} -\sin \theta(s) \\ \cos \theta(s) \end{bmatrix}. \quad (2.49)$$

## 2.6 Загасання

Як показано в попередніх главах, загальне рішення для акустичного тиску можна записати у вигляді:

$$P(r, \omega) = Ae^{-i\omega t}, \quad (2.50)$$

де

$$A = \frac{1}{4\pi} \sqrt{\frac{c_0(s) \cos \theta(0)}{c_0(0) S q(s)}} \exp \left[ -i\omega \frac{1}{2} \frac{p(s)}{q(s)} n^2 \right], \quad (2.51)$$

для апроксимації променів Гауса. Будь-який вираз не враховує розсіювання енергії через поглинання матеріалу і через передачу енергії щоразу, коли акустична хвиля відскакує на межі. Таке розсіювання залежить від частоти (воно більш актуальне в міру збільшення частоти) і передбачає використання скорегованої амплітуди  $a$ , що відповідає вихідній амплітуді, помноженій на два коефіцієнти затухання  $\varphi_r$  і  $\varphi_V$ :

$$a = A \cdot \varphi_r \cdot \varphi_V, \quad (2.52)$$

де  $\varphi_r$  і  $\varphi_V$  стоять, відповідно, для розпаду через граничні відбиття та за рахунок поглинання об'єму. Обидва фактори описані далі.

Коефіцієнт затухання  $\varphi_r$  задається виразом:

$$\varphi_r = \prod_{i=0}^{n_r} R_i, \quad (2.53)$$

де  $n_r$  – загальна кількість граничних відбивань;

$R_i$  – коефіцієнт відбиття в  $i$ -му відбитті.

Випадок без жодного відбиття ( $n_r = 0$ ) відповідає  $\varphi_r = 1$ . Взагалі кажучи, межі можуть бути одного з чотирьох типів:

- поглинаючі: енергія хвилі передається повністю до середовища над границею, тому  $R = 0$  і розповсюдження променів. закінчується на межі;
- жорсткі: енергія хвилі відбивається повністю на межі, без зміни фази, тому  $R = 1$ ;
- вакуумні: енергія хвилі відбивається повністю на межі зі зміною фази  $\pi$  радіанів, тому  $R = -1$ ;

- еластичні: енергія хвилі частково відбита,  $R$  має комплексне значення і  $|R| < 1$ .

Взагалі коефіцієнт відбиття є реальним, коли  $\alpha_{cp} = \alpha_{cs} = 0$ , а кут падіння  $\theta$  менший від критичного кута  $\theta_{cr}$ , причому  $\theta_{cr}$ , заданий виразом:

$$\theta_{cr} = \arcsin\left(\frac{c_{pz}}{c_{pz}^1}\right)$$

Більше того, загасання є незначним, коли  $\theta < \theta_{cr}$ , а для малих  $\theta$  енергія, що передається на хвилі зсуву в пружному середовищі, є лише невеликою часткою всієї переданої енергії.

Коефіцієнт  $\varphi_V$  задається спадаючою експонентою:

$$\varphi_V = \exp(-\alpha_T s), \quad (2.53)$$

де  $s$  – довжина дуги;

$\alpha_T$  – коефіцієнт загасання Торпа (Thorp), який залежить від частоти і задається в  $\frac{\text{дБ}}{\text{м}}$ , і дорівнює:

$$\alpha_T = \frac{40f^2}{4100 + f^2} + \frac{0.1f^2}{1 + f^2}, \quad (2.54)$$

де  $f$  – частота, що задана в кГц.

## 2.7 Чисельні методи

В даному підрозділі будуть розглянуті чисельні методи, використані при розробці комп'ютерної системи.

### 2.7.1 Метод Рунге-Кутта-Фелберга

Метод Рунге-Кутта-Фелберга (або метод Фелберга) – це алгоритм в чисельному аналізі для чисельного рішення диференціальних рівнянь. Він був розроблений Ервіном Фелбергом і базується на широкому класі методів Рунге-Кутта. Методи Рунге-Кутти є методами точкових нахилів, але були розроблені для забезпечення підвищеної точності з більшими ступінчастими

розмірами і без необхідності застосування вищих диференціалів (за винятком першої похідної) функції.

Новизна методу Фелберга полягає в тому, що це вбудований метод з сімейства Рунге-Кутта, що означає, що ідентичні оцінки функцій використовуються спільно між собою для створення методів різного порядку та аналогічних констант помилок. Згодом даний метод був названий RKF45 тому, що цей метод четвертого порядку з оцінкою помилок п'ятого порядку. Виконуючи додатковий підрахунок, помилки в рішенні можна оцінити та вести їх контроль, використовуючи вбудований метод вищого порядку, який дозволяє автоматично визначити адаптивний крок [7].

Формули для п'ятого порядку виглядають таким чином:

$$\Delta y_n = \frac{16}{135} k_1 + \frac{6656}{12825} k_3 + \frac{28561}{56430} k_4 - \frac{9}{50} k_5 + \frac{2}{55} k_0, \quad (2.55)$$

$$\begin{aligned} k_1 &= h \cdot f(x_n, y_n), \\ k_2 &= h \cdot f\left(x_n + \frac{1}{4}h, y_n + \frac{1}{4}k_1\right), \\ k_3 &= h \cdot f\left(x_n + \frac{3}{8}h, y_n + \frac{3}{32}k_1 + \frac{9}{32}k_2\right), \\ k_4 &= h \cdot f\left(x_n + \frac{12}{13}h, y_n + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right), \\ k_5 &= h \cdot f\left(x_n + h, y_n + \frac{439}{216}k_1 - 8k_2 + \frac{3860}{513}k_3 + \frac{845}{4104}k_4\right), \\ k_6 &= h \cdot f\left(x_n + \frac{1}{2}h, y_n - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 + \frac{11}{40}k_5\right). \end{aligned} \quad (2.56)$$

Реалізація даного методу знаходиться в класі RayCalculator – метод RKF45(...).

### 2.7.2 Розрахунок рівнянь Ейконала

Рівняння Ейконала (2.25) інтегруються, використовуючи метод типу Рунге-Кутта, відомий як метод Рунге-Кутта-Фельберга (або RKF45). Як і в інших методах Рунге-Кутти, метод RKF45 починається з перезапису



оригінальної системи рівнянь до вигляду лінійного диференціального векторного рівняння:

$$\frac{dy}{ds} = f, \quad (2.57)$$

де:

$$y = \begin{bmatrix} r \\ z \\ \sigma_r \\ \sigma_z \end{bmatrix}, \quad (2.58)$$

та

$$f = \begin{bmatrix} \sigma_r \\ \sigma \\ \sigma_z \\ \sigma \\ \frac{\partial \sigma}{\partial r} \\ \frac{\partial \sigma}{\partial z} \end{bmatrix}. \quad (2.59)$$

На кожному етапі інтеграції метод проходить через набір проміжних етапів. Після кожної інтеграції метод RKF45 пропонує не одне, а два різних рішення. Якщо дельта цих рішень відрізняється більше, ніж певний заданий поріг, то крок  $ds$  променів може бути зменшений в два рази, а розрахунки будуть повторені, забезпечуючи середнє значення для контролю точності дельти. З метою запобігання нескінченному циклу, алгоритм перериває і припиняє обчислення, якщо послідовне порівняння рішень не може зблизитися.

### 2.7.3 Розрахунок точок після відбиття

Моделювання розповсюдження хвилі було б неповним без належного оновлення значень  $p(s)$  та  $q(s)$  після кожного відображення кордону. Беручи за основу метод з бібліотеки Bellhop, який використовується для оновлення значень, рівняння задається:

$$\begin{cases} p' = p + qr_n, \\ q' = q. \end{cases}, \quad (2.60)$$

де  $p$  та  $q$  – значення перед відбиттям;

$p'$  та  $q'$  - значення після відбиття.

Корекція відбиття  $r_n$  розраховується таким чином:

$$r_n = r_m \cdot \frac{4C_n - 2r_m C_s}{c}, \quad (2.61)$$

де  $c$  значення швидкості звуку на межі.

Значення  $r_m$  рахується таким чином:

$$r_m = \frac{T_g}{T_h}, \quad (2.62)$$

Значення  $C_n$ ,  $C_s$ ,  $T_g$  та  $T_h$  рахуються згідно заданим рівнянням:

$$T_g = (\sigma \cdot \tau_b), \quad (2.63)$$

$$T_h = (\sigma \cdot n_b), \quad (2.64)$$

$$c_n = (\nabla c \cdot \sigma_n), \quad (2.65)$$

$$c_s = (\nabla c \cdot \sigma), \quad (2.66)$$

де  $\tau_b$  – відповідає граничній дотичній;

$n_b$  – нормаль до межі;

$\nabla c$  – градієнт швидкості звуку;

$\sigma$  – звукова повільність;

$\sigma_n$  – нормаль до звукової повільності.

Параметри  $\sigma_n$  та  $\sigma$  обчислюються в точці відбиття та дорівнюють:

$$\sigma = \frac{1}{c} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, \quad (2.67)$$

$$\sigma_n = \frac{1}{c} \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}. \quad (2.68)$$

#### 2.7.4 Корекція за рахунок явища рефракції

Для розрахунку корекції явищем рефракції у бібліотеці Bellhop була використана корекція заломлення, яку можна записати як:

$$p' = p + q\tilde{r}_n, \quad (2.69)$$

де  $p'$  – скореговане значення.

В свою чергу  $\tilde{r}_n$  дорівнює:

$$\tilde{r}_n = -\tilde{r}_m \frac{2c_{nj} - \tilde{r}_m c_{sj}}{c}, \quad (2.70)$$

де :

$$\tilde{r}_m = \frac{\sigma_n(s)}{\sigma_z(s)}, \quad (2.71)$$

$$C_{nj} = \delta \nabla c \cdot \sigma_n, \quad (2.72)$$

$$C_{sj} = \delta \nabla c \cdot \sigma, \quad (2.73)$$

де  $c$  – швидкість звуку в положенні прибуття;

$\delta \nabla c$  – варіація  $\nabla c$  між початковою та кінцевою позиціями.

#### 2.7.5 Відбиття променів від межі

Закони дзеркального відбивання просто стверджують, що кут падіння повинен бути рівний куту відбиття і що падаючий промінь разом із нормаллю поверхні та відбитим променем повинен лежати у загальній площині. Реалізація такого висловлювання з точки зору кутів може стати досить неефективною, якщо пам'ятати, що межа не обов'язково є такою. У цьому випадку, для обчислення кутів падіння та відбиття необхідні обчислення променевих та граничних нахилів, які розраховуються, використовуючи зворотні тригонометричні функції, щоб отримати кути променю чи межі відносно горизонталі, і визначити, як їх поєднувати. Такий підхід може стати

досить громіздким. Ефективний метод обчислення відбиття променів полягає у використанні виразу:

$$e'_s = e_s + (2 \cos \theta_1) n_b, \quad (2.74)$$

де  $n_b$  – нормаль до межі в точці падіння;

$e'_s$  – значення після відбиття;

$\cos \theta_1$  – добуток  $(-e_s)$  на  $n_b$ .

Таким чином, кожного разу, коли промінь перетинає кордон, алгоритм обчислює точку перетину кордону, яка стає точкою падіння, і координати точки використовуються для обчислення нормалі межі, і застосовується рівняння (2.74). Інтеграція рівняння Ейконала потім перезапускається в точці падіння (як би джерело тепер знаходилось на межі), при цьому кут запуску визначається компонентом  $e'_s$ .

Висновки до розділу

У даному розділі розглянуті теоретичні та чисельні методи, які стали основою для модулів комп'ютерної системи, що створюють модель трасування променів і імітують результати, які можна було б отримати тільки в реальних умовах. Дані методи були запропоновані та реалізовані в бібліотеках Bellhop та TRACEO, можливості яких описані в розділі 1.2.

### 3. РОЗРОБКА КОМПОНЕНТІВ КОМП'ЮТЕРНОЇ СИСТЕМИ

Архітектура усіх сучасних систем передбачає розробку модулів програми, що мають послідовну залежність, тобто результати роботи попереднього модуля використовуються наступним. У цьому розділі будуть розглянуті основні модулі, які виконують певне функціональне призначення і складаються з менших модулів.

#### 3.1 Особливості реалізації та використання додаткових модулів

Основною причиною використання фреймворку Qt була вбудована система колбеків (зв'язок за допомогою зворотних викликів) на якому побудована основна частина взаємозв'язку архітектури системи. У Qt присутня альтернатива техніці зворотного виклику, яка називається сигнали та слоти. Сигнал випромінюється, коли відбувається певна подія чи коли він був навмисно викликаний, а слот, в свою чергу – функція, яка викликається у відповідь на певний сигнал. Слоти можна використовувати не лише для прийому сигналів, а також як звичайні функції. Подібно до того, як об'єкт не знає, чи отримує щось сигнали, так і слот не знає, чи є пов'язані з ним якісь сигнали. Це гарантує, що за допомогою Qt можна створити насправді незалежні компоненти. Даний механізм можна використовувати лише для класів, які успадковують QObject, або один з його підкласів (наприклад QWidget).

Великою перевагою сигналів та слотів над реалізацією в бібліотеці Boost є те, що механізм колбеків є безпечнішим. Параметри сигналу мають відповідати параметрам слотів за типом (хоча допускається те, що слот має меншу кількість параметрів, які може прийняти, ніж у сигналі, який він отримує, оскільки ігнорує зайві аргументи). Оскільки сигнатура сумісна, компілятор може виявити невідповідність типів. Синтаксис SIGNAL і SLOT на основі рядків виявить невідповідність типу під час виконання. Сигнали та слоти зв'язані між собою. Тобто клас, який видає сигнал, не знає і не

підключається, які слоти приєднують сигнал. Сигнали та слоти можуть приймати будь-яку кількість аргументів будь-якого типу. Вони повністю безпечні.

Даний механізм дає змогу підключити стільки сигналів, скільки потрібно підключити до одного слоту, і сигнал можна підключити до стільки слотів, скільки буде потрібно. Можна навіть підключити сигнал безпосередньо до іншого сигналу, що призведе до того, що при спрацюванні першого буде негайно викликано приєднаний до нього сигнал. Разом сигнали та слоти складають дуже потужний механізм проектування компонентів та міжкомпонентного зв'язку.

В багатьох компаніях використовують свої внутрішні інструменти або бібліотеки, що полегшують створення та проектування ПЗ. Допоміжний модуль – це компонент з деяким набором інструментів, який полегшує та пришвидшує процес проектування, створення та вдосконалення продукту. В даній бібліотеці знаходяться інструменти для роботи не тільки міжпоточної, а й міжкомп'ютерної передачі даних за допомогою сокетів. Але для рішення задачі моделювання розповсюдження звукових хвиль в гідросфері було взято декілька складових даної бібліотеки, такі як: макроси та кілька модулів для розробки візуального інтерфейсу в середовищі QtQuick, яка використовує декларативну мову QML.

В Qt існує дуже зручний макрос – `Q_PROPERTY`, який дозволяє додавати будь-яку кількість властивостей до будь-якого класу `QObject`. Але в деяких випадках цим доволі незручно користуватись. Для створення потрібно написати код виду (рис. 3.1.). Тут «property» є властивістю, «getProperty» – метод для читання значення властивості, «setProperty» – метод задання значення властивості, а «propertyChanged» – сигнал, який відправляється після того, як значення властивості буде змінено. `Q_PROPERTY` взагалом використовується для того, щоб зробити властивість в C++ частині доступною для QML частини. Якщо з'явилась потреба змінити властивість з C++, то для цього використовується метод «setProperty» для того, щоб QML частина

дізналась про зміни, адже при використанні «setProperty» буде надіслано сигнал «propertyChanged», після чого властивість в QML частині змінить своє значення.

```

1  #ifndef CLASS_H
2  #define CLASS_H
3
4  #include <QObject>
5
6  class Class : public QObject
7  {
8      Q_OBJECT    //або Q_GADGET
9      Q_PROPERTY(int property READ getProperty WRITE setProperty NOTIFY propertyChanged)
10 public:
11     explicit Class(QObject *parent = nullptr);
12     /* можлива реалізація setterів та getterів
13      *
14      *      Q_INVOKABLE int getSomeProperty()const
15      *      {
16      *          return property;
17      *      }
18      *      Q_INVOKABLE void setSomeProperty(const int &value)
19      *      {
20      *          property = value;
21      *          emit propertyChanged();
22      *      }
23      */
24 private:
25     int property;
26
27 signals:
28     void propertyChanged();
29
30 public slots:
31     int getSomeProperty()const
32     {
33         return property;
34     }
35     void setSomeProperty(const int &value)
36     {
37         property = value;
38         emit propertyChanged();
39     }
40 };
41
42 #endif // CLASS_H
43

```

Рисунок 3.1 – Приклад написання Q\_PROPERTY звичайним шляхом

Даний спосіб не досить складний, але в ньому є декілька недоліків, наприклад:

- при створенні нової властивості наступним кроком обов’язково потрібно задати їй значення, адже, якщо створити властивість з

типом показника на екземпляр класу, то початковим значенням після створення буде сміття, а не значення nullptr чи коректне значення;

- після додавання потрібна реалізація усіх методів.

Тому були використані макроси (рис. 3.2). Їх використання зводиться лише до одної строки коду (рис. 3.3) Вони позбавлені усіх вищезазначених недоліків і дозволяють поліпшити не лише читабельність коду, а й швидкість розробки та проектування.

```
1  #pragma once
2
3  #define AUTO_PROPERTY(TYPE, NAME, GETTER, SETTER, NOTIFIER, DEFAULT) \
4      Q_PROPERTY(TYPE NAME READ GETTER WRITE SETTER NOTIFY NOTIFIER) \
5      public: \
6          TYPE GETTER() const { return this->m_ ## NAME; } \
7          void SETTER(TYPE value) { \
8              if (this->m_ ## NAME == value) return; \
9              this->m_ ## NAME = value; \
10             emit NOTIFIER(value); \
11         } \
12         Q_SIGNAL void NOTIFIER(TYPE value); \
13     private: \
14         TYPE m_ ## NAME = DEFAULT;
15
16 #define AUTO_PROPERTY_WITH_PRIVATE_SETTER(TYPE, NAME, GETTER, SETTER, NOTIFIER, DEFAULT) \
17     Q_PROPERTY(TYPE NAME READ GETTER WRITE SETTER NOTIFY NOTIFIER) \
18     public: \
19         TYPE GETTER() const { return this->m_ ## NAME; } \
20         Q_SIGNAL void NOTIFIER(TYPE value); \
21     private: \
22         TYPE m_ ## NAME = DEFAULT; \
23         void SETTER(TYPE value) { \
24             if (this->m_ ## NAME == value) return; \
25             this->m_ ## NAME = value; \
26             emit NOTIFIER(value); \
27         }
28
29 #define AUTO_FLOAT_PROPERTY(TYPE, NAME, GETTER, SETTER, NOTIFIER, DEFAULT) \
30     Q_PROPERTY(TYPE NAME READ GETTER WRITE SETTER NOTIFY NOTIFIER) \
31     public: \
32         TYPE GETTER() const { return this->m_ ## NAME; } \
33         void SETTER(TYPE value) { \
34             if (qFuzzyCompare(this->m_ ## NAME, value)) return; \
35             this->m_ ## NAME = value; \
36             emit NOTIFIER(value); \
37         } \
38         Q_SIGNAL void NOTIFIER(TYPE value); \
39     private: \
40         TYPE m_ ## NAME = DEFAULT;
41
42 #define AUTO_FLOAT_PROPERTY_WITH_PRIVATE_SETTER(TYPE, NAME, GETTER, SETTER, NOTIFIER, DEFAULT) \
43     Q_PROPERTY(TYPE NAME READ GETTER WRITE SETTER NOTIFY NOTIFIER) \
44     public: \
45         TYPE GETTER() const { return this->m_ ## NAME; } \
46         Q_SIGNAL void NOTIFIER(TYPE value); \
47     private: \
48         TYPE m_ ## NAME = DEFAULT; \
49         void SETTER(TYPE value) { \
50             if (qFuzzyCompare(this->m_ ## NAME, value)) return; \
51             this->m_ ## NAME = value; \
52             emit NOTIFIER(value); \
53         }
```

Рисунок 3.2 – Макроси для полегшення використання Q\_PROPERTY



Рисунок 3.3 – Використання макроса з додаткового модуля

### 3.2 Головний модуль

Головний модуль містить точку входу в програму і виконує такі функції:

- регулює злагоджену роботу усіх модулів програми;
- виконує завантаження попередньо згенерованих даних у буфер, який використовується на протязі усієї роботи програми, або може бути вимкненим, при потребі, користувачем;
- при знаходженні помилки в графічному відображенні, дозволяє згенерувати новий буфер в мультипоточному режимі;
- оброблює дані, що приходять з графічного інтерфейсу користувача та на основі результатів виконує дії, які запросив користувач;
- при потребі може генерувати звуки гвинтів підводного човна, якщо його координати знаходяться в межах зони освітленості.

Його інтерфейс доволі простий і в більшості методи головного модуля використовуються для взаємодії з графічним інтерфейсом користувача. Екземпляр даного модуля є singleton, тому він може бути доступним з будь-якої області програми завдяки методу getInstance(), який повертає вказівник на статичний екземпляр класу RayTracer.

Щодо методів класу RayTracer – їх набір досить малий, але цього достатньо для злагодженої роботи усієї системи. При створенні екземпляру класу відбувається ініціалізація усіх модулів та завантаження з файлів, з розширенням .json, попередньо згенерованих даних, які на протязі усієї роботи програми знаходяться у буфері. Завантаження, як і створення цього буферу, відбувається в мультипоточному режимі за допомогою вбудованого класу QFuture та QFutureWatcher, які виконують обчислення в окремому потоці та повертають результати виконання. Дана реалізація сприяє поліпшенню швидкості завантаження або генерування даних.

Використання попередньо згенерованих даних зумовлено тим, що в інтегрованому режимі (коли комп'ютерна система є складовою спеціального автоматизованого комплексу), потрібна висока швидкодія і без їх використання автоматизований комплекс працював би ненадійно (адже для генерації даних потрібно від 0.2с до 1.3с в залежності від складності обчислень та дистанції) через те, що дані не встигали б обчислюватись.

Робота RayTracer починається з виклику методу start(). В даному методі створюється екземпляр модулю графічного зображення результатів симуляції та відбувається ініціалізація обробника сигналу зміни відображення станції з «МГК-335. Платина» на «МГК-345. Бронза» і навпаки. Роботу даних станцій виконує модуль симуляції розповсюдження звукових хвиль окремо для кожної.

Головний модуль, перш за все, створювався як підпрограма, яка надає близький до реального досвід роботи на гідроакустичній станції, тому важливою задачею є його інтеграція до більш складного програмного комплексу, а саме підключення самого модулю та сторонньої бібліотеки побудови графіків.

### 3.2.1 Підключення сторонньої бібліотеки побудови графіків

Головний модуль хоч і є точкою входу в програму, але для взаємодії з ним потрібно попередньо створити віджет (що виконується достатньо легко і не потребує пояснень), а вже в ньому зробити декілька маніпуляцій. Для початку потрібно інтегрувати додаткову бібліотеку QCustomPlot [8], що займається візуалізацією відображення роботи програми у вигляді графіків. Підключення бібліотеки виглядає таким чином:

1. Після створення проекту QtWidgets в папку з вхідним кодом завантажуюмо два файли бібліотеки (рис. 3.3).
2. Надалі, ці файли потрібно додати до проекту. Для цього в вікні QtCreator натискаємо правою клавішею на проект та обираємо пункт

«Додати існуючі файли». Обираємо два файли і вони будуть додані до проекту.

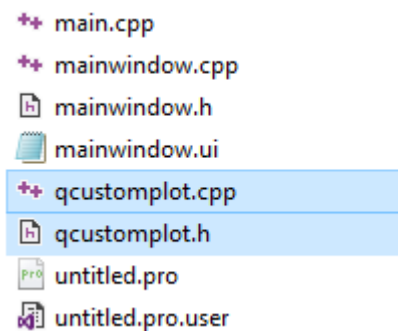


Рисунок 3.3 – Додані файли бібліотеки до вхідного каталогу

3. Перейдемо до файлу проекту з розширенням .pro та додамо слово «printsupport», як це зображено на рис. 3.4.

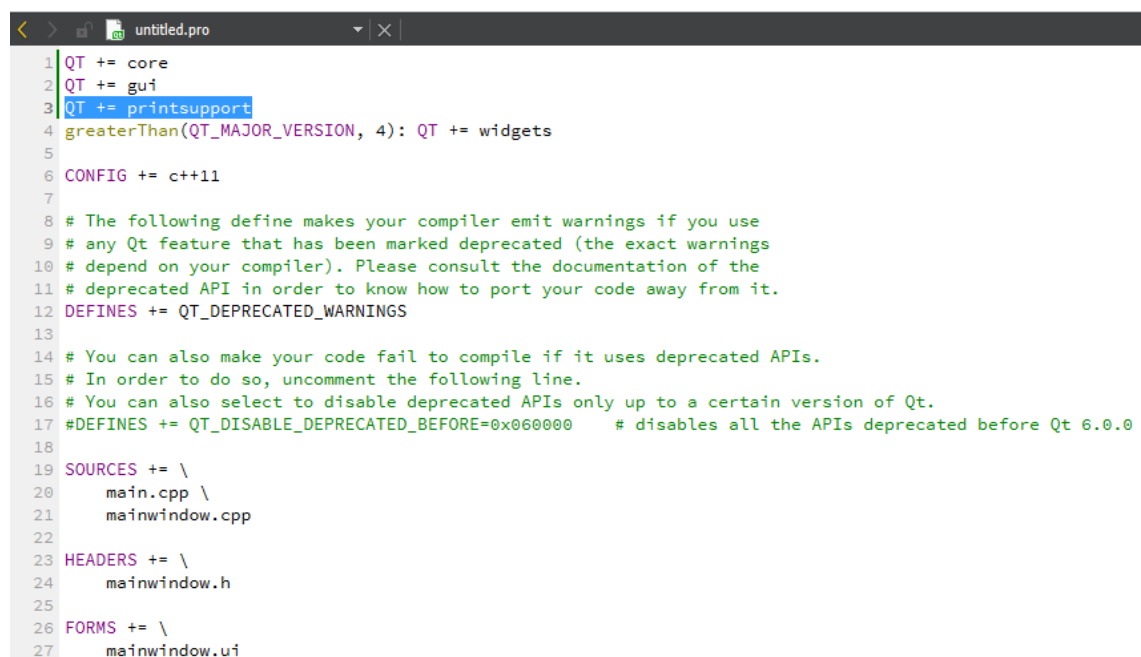


Рисунок 3.4. Додавання «printsupport» до файлу проекту

4. Тепер потрібно перейти до редактору форм та створити вікно віджету. Для цього достатньо двічі натиснути на файл з розширенням .ui і відкриється вікно дизайнеру. Далі, потрібно перенести Widget з

панелі віджетів до вікна та задати йому розмір. Після цього потрібно натиснути правою кнопкою миші та обрати пункт «Перетворити в». У вікні, що з'явилося потрібно додати клас графіків QCustomPlot та натиснути кнопку підтвердження. Ззовні нічого не змінилось, але, якщо запустити програму, то у вікні з'явиться координатна сітка (рис 3.5).

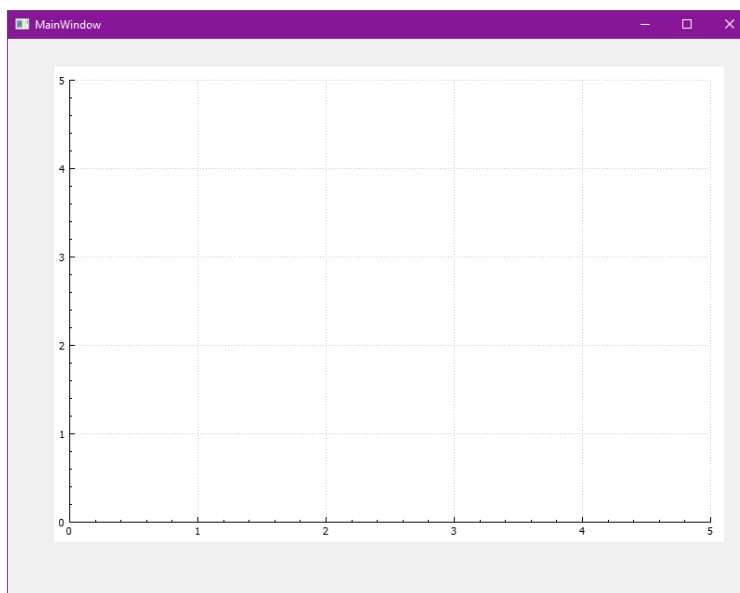


Рисунок 3.5 – Координатна сітка після підключення QCustomPlot

### 3.2.2 Підключення головного модулю

Після вдалого налагодження потрібно додати ще декілька віджетів в дизайнері:

1. Створити віджет та повторити пункт 4 для нього.
2. Додати QQuickWidget для того, щоб відображались кнопки та важелі для управління програмою.

В завершенні розмістити вікна віджетів в будь-якому зручному порядку і приступати до підключення головного модулю. Для роботи головного модулю треба створити екземпляр класу у віджеті та передати за допомогою методу «start» передати вказівники на вікна віджетів типу QCustomPlot та

задати вхідний файл для вікна віджету QQuickWidget разом з контекстною змінною, яка вказує на клас головного модулю, як на рис. 3.6.

```
17  Widget::Widget(QWidget *parent) :  
18      QWidget(parent),  
19      ui(new Ui::Widget)  
20  {  
21      ui->setupUi(this);  
22  
23      //createBuffer();  
24      //loadBuffer();  
25  
26      RayTracer *rayTracer = RayTracer::getInstance();  
27      rayTracer->start(ui->w_ssp, ui->w_rays);  
28      ui->quickWidget->setSource(QUrl("../ControlPanel.qml"));  
29      ui->quickWidget->rootContext()->setContextProperty("rayTracer", rayTracer);
```

Рисунок 3.6 – Код додавання головного модулю до віджету

### 3.3 Модуль симуляції розповсюдження звукових хвиль

Основна задача даного модулю – генерування та зберігання результатів розрахунків програми для подальшої їх обробки та виведення за допомогою модулю графічного зображення симуляції результатів симуляції. В даному підрозділі будуть переглянуті усі підмодулі, їх реалізація та задачі, які вони виконують.

#### 3.3.1 Модуль розрахунків симуляції

Основна функція головного модулю симуляції – це імітація роботи гідроакустичної станції в максимально реальних умовах з урахуванням ВРШЗ. Функціонал даного модулю виконується класом Simulation. В результаті того, що на даний модуль припадає дуже велика низка функціоналу, його мав би розтягнутися більше ніж на 20 методів, але кількість методів не перевищує 14, як і кількість властивостей, описаних за допомогою макросів. При підрахунках гетери та сетери не враховувались.

Робота даного модулю починається з виклику методу start(). Даний виклик ініціалізує обробника сигналів, таких як:

- сигнал натиску на кнопку «Перерахувати» в графічному інтерфейсі;
- сигнал зміни типу гідрології;
- сигнал зміни стану відбиття підводного човна.

Окрім цього, ще викликаються методи `calculate()`, `calculateSpeed()` та `calculateWave()`.

Метод `calculateSpeed()` викликається першим, адже він використовується для того, щоб згенерувати данні одного з семи типу гідрології, і ці данні потрібні для наступних розрахунків. При первісному проектуванні архітектури, реалізація цього методу знаходилась у `calculate()`, але для поліпшення читабельності коду було вирішено винести її в окремий метод.

Метод `calculateWave()` використовується для завантаження даних про конкретну глибину випромінення поточного типу гідрології з буферу. Це набагато пришвидшує роботу системи. Але при вимкненні буферу користувачем даний метод не викликається і всі розрахунки відбуваються в реальному часі.

Метод `calculate()` створює екземпляр модулю розрахунків траєкторії розповсюдження звукового променя та ініціалізує низку таких параметрів:

- встановлює максимальну дистанцію розповсюдження звукового променя;
- встановлює максимальну глибину зони розповсюдження;
- задає параметри зоні розповсюдження;
- встановлює частоту звукової хвилі;
- встановлює профіль розповсюдження на основі даних, розрахованих за допомогою `calculateSpeed()`;
- створює поверхню дна;
- встановлює кількість променів.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

43

Після ініціалізації, в залежності від того, чи увімкнений буфер, виконуються розрахунки. Якщо буфер увімкнений, то потреба в розрахунках відпадає і викликається метод `calculateWave()`, а якщо ні – виконуються для кожного променя окремо, після чого результат зберігається в тимчасовому буфері. Окрім цих методів є ще наступні:

- `interpolate(vector x, vector y, double point, bool extrapolate)` використовується для лінійної інтерполяції та екстраполяції. Дана функція приймає: вектор довжин по осі X; вектор довжин по осі Y; значення на осі X, для якого потрібно знайти значення та булеву змінну, яка регулює чи використовується екстраполяція, чи ні. Результатом повернення функції є результат з плаваючою комою, що дорівнює шуканому значенню;
- `calculateBackWave()` створений для розрахунку відбитої від підводного човна хвилі. Довжина вектору результату напряму дорівнює горизонтальній дистанції від випромінювача (координата дорівнює нулю) до розміщення підводного човна. Беручи до уваги, що даний метод буде достатньо часто викликатись, було вирішено, що для відображення будуть використовуватись буферизовані дані, які завантажувались з файлів, для поліпшення швидкості відгуку на кожний виклик методу. При виході підводного човна з місцезнаходження зони освітленості, усі попередньо згенеровані дані, що знаходяться в тимчасовому буфері будуть видалені;
- `constrainVector(double start, double end, int size)` повертає вектор розмірності `size` з початковим значенням `start`, кінцевим значенням `end` та низкою проміжних значень, які відрізняються на певний крок один від одного, починаючи зі `start + step` закінчуючи `end – step`;
- `findCoordinate(double value, vector vec)` використовується для того, щоб отримати позицію значення наближеного до `value` у векторі `vec`.

Використовується лише для вектору, в якому шукане значення є наближено або повністю відповідним;

- `reflectionBottomPoint(vector points, double reflDepth, int number)` використовується для знаходження позиції в векторі `points` останнього наближеного або точного значення `reflDepth`. Якщо `number` дорівнює одиниці, то буде шукати другий за чергою, тобто передостаннього. У разі не знаходження – повертає -1;
- `reflectionTopPoint(vector points, double reflDepth, int number)` використовується для знаходження позиції в векторі `points` останнього точного значення `reflDepth`, яке за замовчуванням дорівнює нулю. Якщо `number` дорівнює одиниці, то функція поверне позицію другого з кінця елементу. Якщо такого елементу не існує – поверне -1;
- `isReflected(double distance, double depth)` використовується для визначення, чи знаходиться координати підводного човну в зоні освітленості, для поточно вирахованих даних переміщення звукових променів, чи ні. Також, даний функція використовує метод `findCoordinate` для полегшення знаходження позиції у тимчасовому буфері. Алгоритм даного методу зображено у додатку ІАЛЦ.045490.007 ДЗ;
- `isRecieved(double distance, double depth)` працює за тим самим принципом, але бере дані, що були згенеровані викликом методу `calculateBackWave()`.

### 3.3.2 Модуль звукового профілю

Звуковий профіль, один з найголовніших модулів, адже за допомогою нього задаються параметри, які будуть використовуватись під час всієї роботи системи, а саме:

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

45



- вектор точок профілю, які використовуються при розрахунках імітації явища рефракції;
- коефіцієнт повільності, як для горизонтальної дистанції, так і для глибини. Дані коефіцієнти грають велику роль в загасанні коливань звукових хвиль;
- перша похідна від першої точки вектору профілю.

До функціональних можливостей ще можна віднести декілька методів:

- `init(double step, vector ssp)` виконує ініціалізацію профілю. Приймає число, яке вказує на крок, з яким були згенеровані данні у векторі `ssp`, та сам вектор `ssp`;
- `getSpeed(double z)` повертає приблизне значення швидкості, яке знаходиться між попереднім та наступним значенням в векторі значень звукового профілю, що є наближеними до значення `z`;
- `getDerivative(double z)` повертає похідну швидкості від значення `z`, спираючись на данні вектору звукового профілю;
- `calcAll(double z)` виконує ініціалізацію коефіцієнтів повільності, розрахунок першої похідної першого елементу вектору значень звукового профілю та швидкість для поточної глибини, яку було передано параметром `z`;
- `getSpeed(int i)` повертає значення швидкості з `i`-ої позиції у векторі точок звукового профілю.

### 3.3.3 Модуль задання значень за замовчуванням

Використання даного модулю зумовлено тим, що з'явилась потреба при тестуванні та налаштуванні коефіцієнтів та точок, завдяки яким будувався профіль вертикального градієнту швидкості звуку. Функції даного модуля виконує клас `Munc`. Його назва зумовлена тим, що звуковий профіль, який будується даним модулем, має таку саму назву. Даний звуковий профіль

відображає ідеальні умови для розповсюдження звукових променів. Формула розрахунку даного профілю має такий вигляд:

$$c(z) = 1500 \times \left( 1 - \varepsilon \times \left( \tilde{z} + 1 - (e^{-\tilde{z}}) \right) \right), \quad (3.1)$$

де величина  $\varepsilon$  дорівнює

$$\varepsilon = 0.00737,$$

масштабована глибина  $\tilde{z}$  знаходиться за формулою:

$$\tilde{z} = \frac{2 \times (z - c^1)}{c^1}, \quad (3.2)$$

де  $c^1$  – швидкість звуку, яка була підібрана напротязі тестування і відрізняється від значення запропонованого в джерелі [9];

$z$  – і-та глибина.

На значенні 1500 можна побачити ідеальні умови розповсюдження звукових хвиль в гідросфері для даної комп'ютерної системи (рис. 3.7).

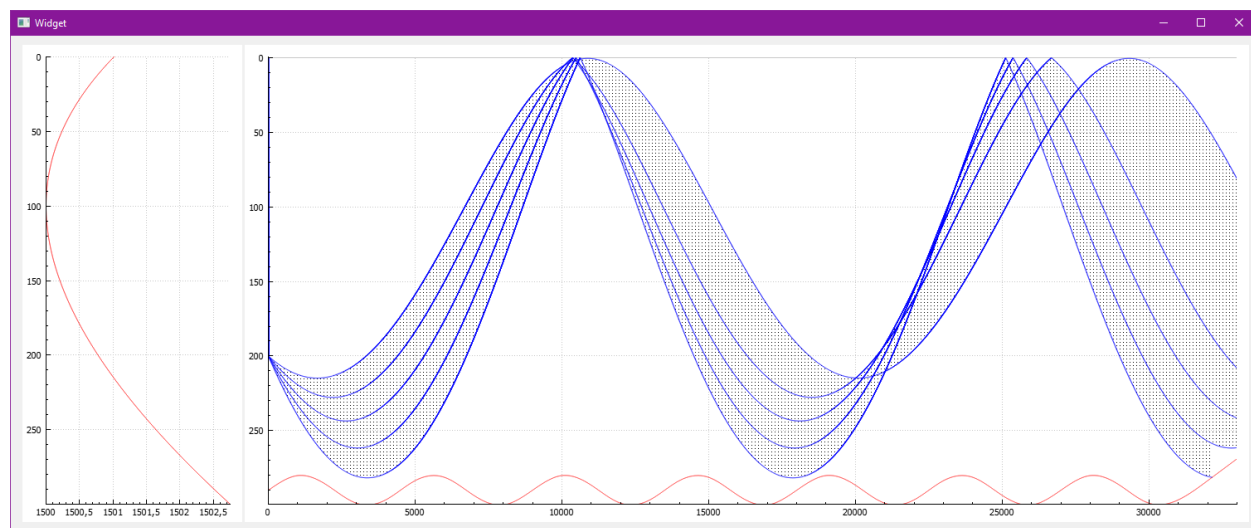


Рисунок 3.7 – Розповсюдження звукових хвиль з профілем Munc

### 3.3.4 Модуль рельєфу

Модуль рельєфу також відіграє важливу роль, адже, при проходженні усього шляху, промені можуть дотикатися до граничних областей, таких як: дно та поверхня водного середовища. Функціонал даного модулю виконує

клас Bathymetry. Даний клас зберігає коефіцієнти, що використовуються при підрахунках частоти звукового променю, та екземпляр класу Spline, який виконує операцію інтерполяції та екстраполяції. Окрім цього, модуль рельєфу має два методи:

- `init(vector distance, vector surface)` ініціалізує початкові параметри для Spline, а саме вектор значень дистанції та вектор значень поверхні рельєфу дна;
- `getSrfcPoint(double r)` викликає внутрішній метод класу Spline, який повертає, розраховане внутрішніми методами інтерполяції, значення.

### 3.3.5 Модуль основних даних

Даний модуль використовується для збереження даних, які були ініціалізовані ще на початку, при створенні модулю розрахунків траєкторії розповсюдження променів. Він зберігає такі дані:

- кількість звукових променів;
- глибину випромінювання;
- частоту звукового променю;
- кут між променями.

### 3.3.6 Модуль розрахунків траєкторії розповсюдження променів

Усі основні розрахунки, що виконуються для отримання результатів траєкторії звукового променю виконуються в даному модулі. Попередньо розглянуті модулі підпорядковуються і використовуються даним модулем. Функціонал даного модулю виконує клас RayCalculator. Даний клас містить в собі методи, для розрахунку трасування променів, що були наведені у другому розділі (розділ 2).

В даному класі представлені такі методи:

- `addPressureCalcZone(...)` використовується для ініціалізації зони розрахунків. Приймає початкові точки X та Y, а ще ширину та висоту зони розрахунків;
- `update(...)` приймає старі та нові значення. Використовується для зберігання нових розрахунків у комірку старих розрахунків і збереження старих до вектору результатів, дані якого будуть відображені в модулі зображення графічних результатів;
- `calcReflectoin(...)` використовується для розрахунку нових точок після відбиття від межі та нового стартового кута нахилу променю до горизонту;
- `rkf45(...)` реалізує алгоритм Рунге-Кутта-Фелберга;
- `rkf45WithPrecision(...)` реалізує алгоритм з адаптивним кроком для алгоритму Рунге-Кутта-Фелберга;
- `calcReflCoeff(double * yOld)` підраховує коефіцієнт, який впливає на швидкість загасання променю;
- `calcRayAmplitude()` використовується для розрахунку амплітуди для поточної координати променю, що в подальшому впливає на його розповсюдження;
- `calcRay(double theta)` реалізує алгоритм, що виконує розрахунки точок звукового променю (рис. 3.8) .

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

49

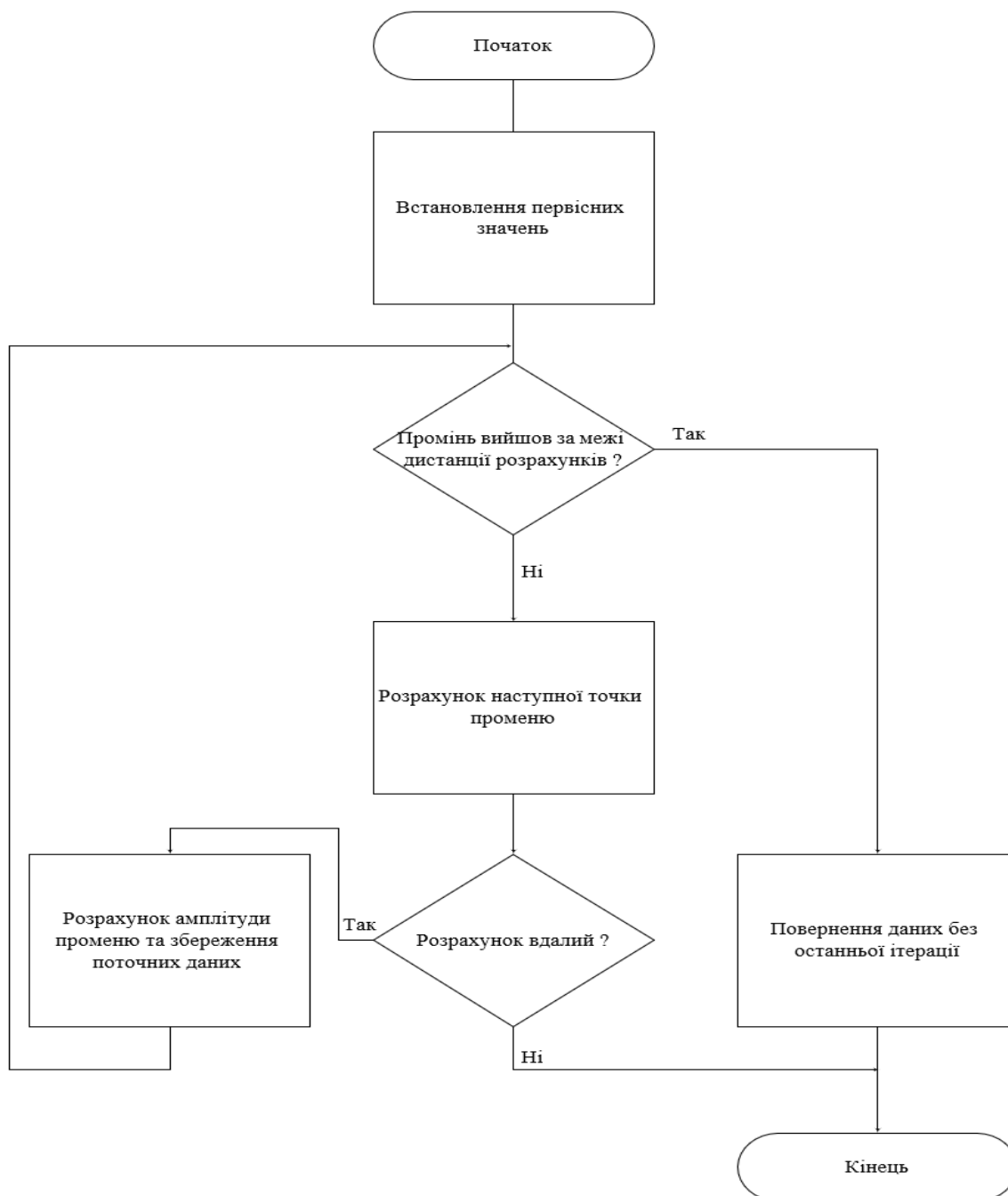


Рисунок 3.8 – Схема алгоритму розрахунку трасування променя

### 3.4 Модуль графічного зображення результатів симуляції

Функції даного модулю реалізує клас `SimulationGraphic`. Даний клас використовує можливості додаткової сторонньої бібліотеки `QCustomPlot` в повному обсязі. Після створення екземпляру класу в головному модулі системи, відбувається копіювання даних (згенерованих головним модулем

симуляції), запуск нескінченного таймеру та ініціалізація обробників сигналів. Створюються два обробники:

- першим є обробник сигналу спрацювання таймеру (даний сигнал спрацьовує кожні 500 мс) для того, щоб оновлювати поточний стан результатів, зображених на графіках, а також для відображення постійного переміщення координат підводного човна (якщо швидкість не дорівнює нулю);
- другим є обробник сигналу з головного модуля симуляції, який сповіщає про те, що стан відбиття змінився (тобто потрібно відобразити дані, згенеровані методом `calculateBackWave()` класу `Simulation`.

Окрім цього даний клас має ще й такі методи:

- `paint()` використовується для відображення результатів розрахунків, зроблених в класі `Simulation`. Ще до цього виконує встановлення параметрів відображення графіків: кольорів, візерунків зафарбування, масштаб тощо;
- `paintSpeed()` використовується для відображення на графічній площині результатів розрахунків, а саме точок, що зображають вертикальний градієнт швидкості звуку;
- `setStation()` викликається для того, щоб змінити поточну станцію на іншу, тобто замінити модуль розрахунків симуляції, який використовувався для «Бронзи» на модуль який імітує роботи «Платини» і навпаки;
- `paintSubmarine()` виконує побудову розмітки пунктирною лінією по осі X та Y до координат підводного човна. При виклику виконується перебудова не всього графіку, а лише частини, з'єднаної з координатами підводного човна;
- `repaint()` є слотом, а отже приєднується до сигналу. Сигнал, до якого він приєднується, приходить з модулю розрахунків симуляції. Після

спрацювання сигналу викликається метод `repaint()`, який містить в собі виклики усіх попередньо описаних методів в цьому пункті, для перебудови графіків.

### 3.5 Модуль графічного інтерфейсу користувача

Графічний інтерфейс один з найважливіших модулів для користувача комп'ютерної системи. Інтерфейс, який зображено на рис 3.9 дає змогу вносити маніпуляції з управління даною системою: від переміщення підводного човна до зміни звукового профілю. Даний модуль, на відміну від інших, написаний на декларативній мові QML.

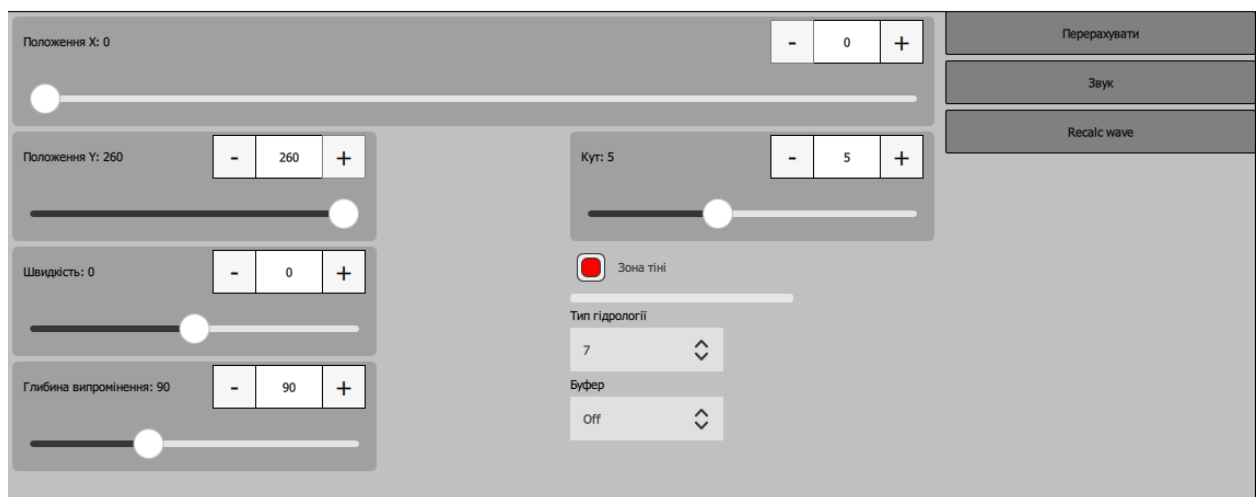


Рисунок 3.9 – Графічний інтерфейс комп'ютерної системи

### 3.6 Загальний алгоритм роботи системи

Комп'ютерна система симуляції розповсюдження звукових променів в гідросфері складається з багатьох компонентів, які взаємозв'язані між собою (рис. 3.11). Кожний з компонентів системи був детально описаний в попередніх підрозділах.

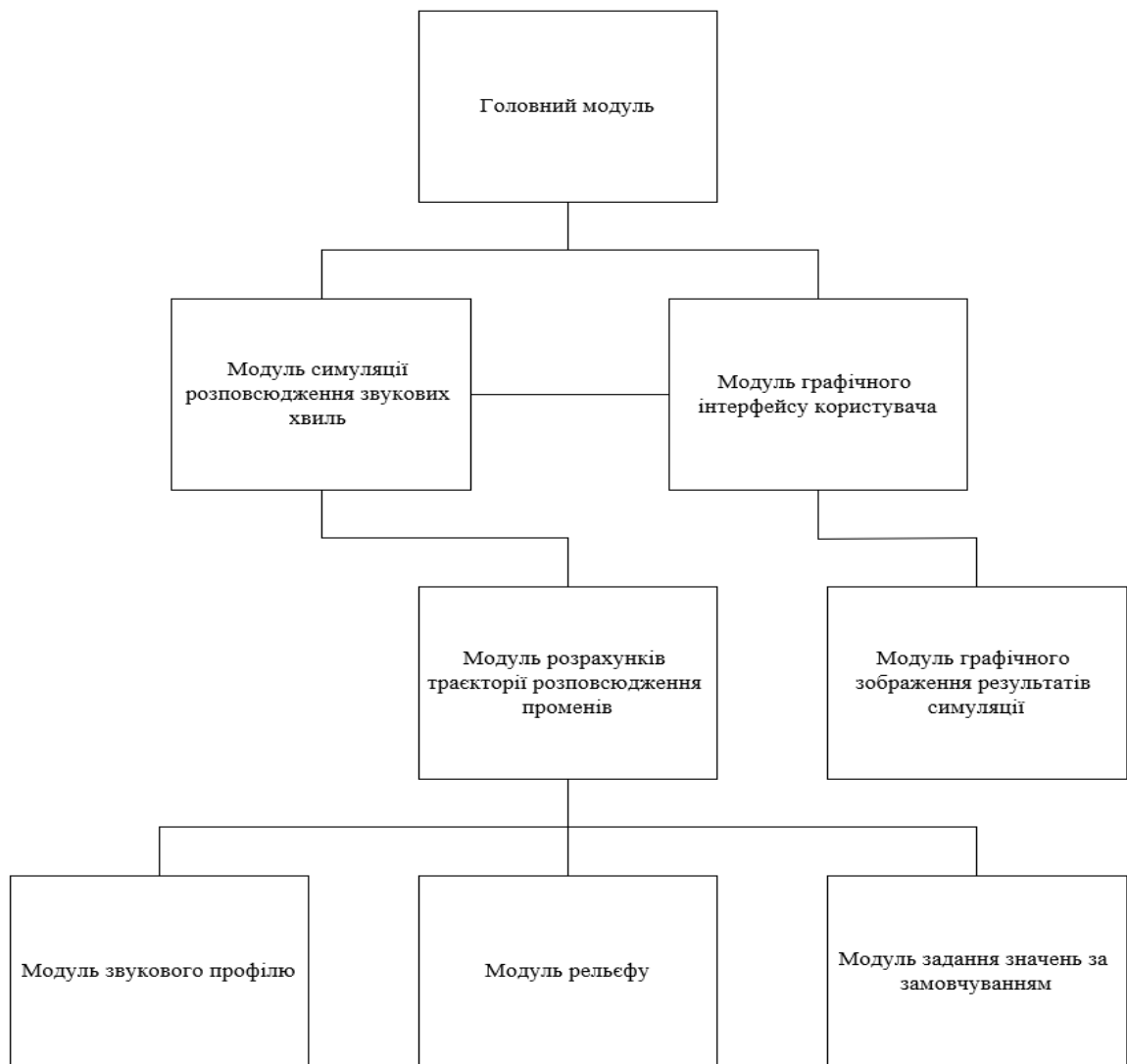


Рисунок 3.11 – Загальна структура системи

#### Висновки до розділу

В даному розділі були розглянуті усі складові комп'ютерної системи. Велику увагу було приділено поясненню особливості їх використання та взаємозв'язку між ними в комп'ютерній системі.



#### 4. КЕРІВНИЦТВО ДЛЯ КОРИСТУВАЧА КОМП'ЮТЕРНОЇ СИСТЕМИ

Розроблена комп'ютерна система розповсюдження звукових хвиль в гідросфері має графічний інтерфейс користувача (рис 4.1). Для взаємодії використовуються важелі та кнопки, за допомогою яких можна змінювати такі параметри:

- горизонтальну дистанцію до підводного човна;
- вертикальну дистанцію до підводного човна;
- горизонтальну швидкість підводного човна (у вузлах);
- кут між променями;
- глибину випромінювання;
- тип гідрології (звуковий профіль).

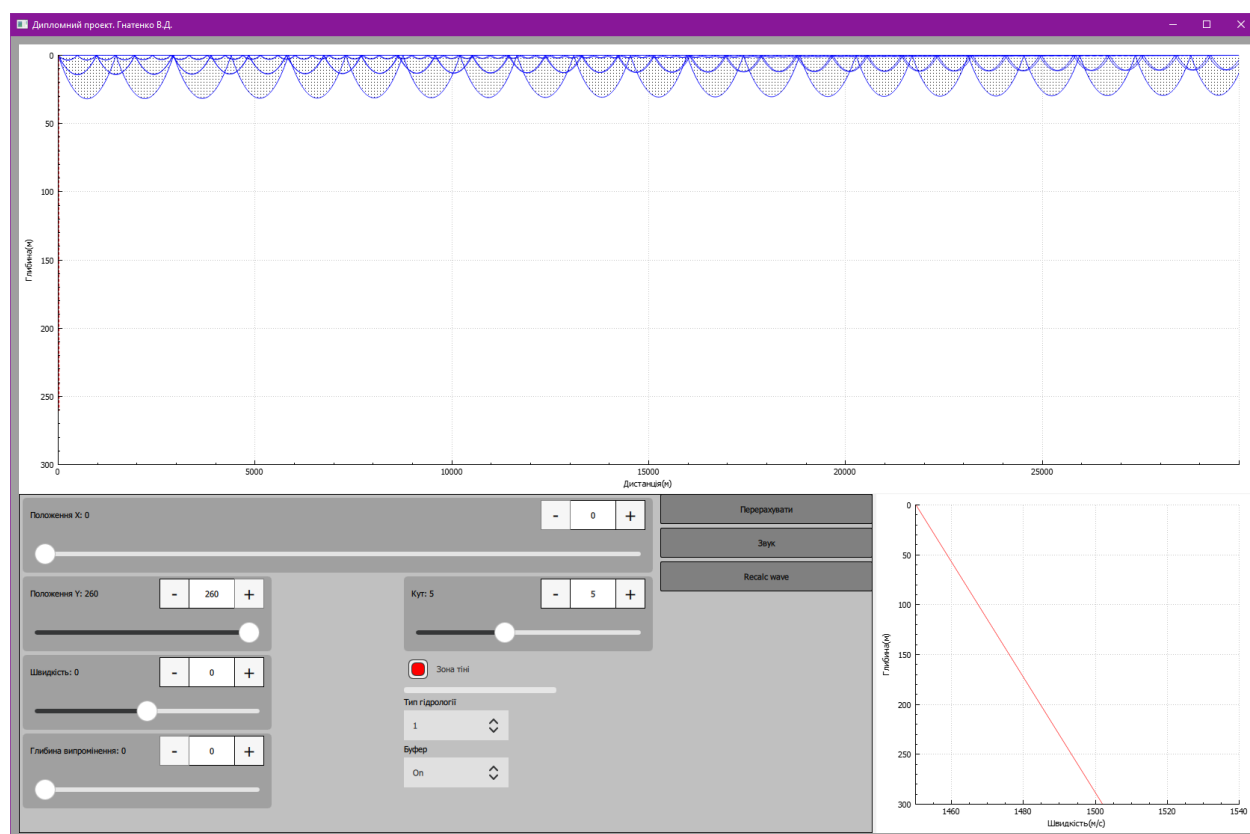


Рисунок 4.1 – Інтерфейс комп'ютерної системи

При зміні положення підводного човна, не залежно від того горизонтального чи вертикального, відносно надводного човна – відбувається

перебудова результатів з урахуванням того, чи є підводний човен в зоні освітленості, чи він в зоні тіні.

При зміні типу вертикального градієнту швидкості звуку, для оновлення розрахунків робити нічого не потрібно, система сама розрахує та поверне результат з новим типом ВРШЗ для поточної глибини випромінення. Також варто зазначити, що зображення типу ВРШЗ знаходиться на графіку в нижній правій області інтерфейсу.

При зміні глибини випромінення система не буде автоматично перебудовуватись, тому після зміни потрібно натиснути кнопку «Перерахувати».

При зміні куту також потрібно натиснути кнопку перебудови результатів. Даний важіль впливає на те, який кут буде між променями і підтримує значення в діапазоні від нуля до одинадцяти.

Переміщення човна задається не тільки важелями положення по горизонталі та вертикалі, а також за допомогою важелю швидкості, який задає значення в діапазоні від -30 до 30 вузлів.

Показчик з назвою «Зона тіні» дає змогу легко побачити, чи є підводний човен, положення якого зображено штрихованою лінією, в зоні тіні. Якщо човен присутній в зоні освітленості, то показчик не буде мати кольору, а якщо ні – буде підсвічуватись червоним кольором. Крім того, смужка знизу показує рівень максимальної амплітуди звукового сигналу, який користувач може почути, якщо буде увімкнений звук.

#### Висновки до розділу

В даному розділі були розглянуті елементи взаємодії з системою, за допомогою графічного інтерфейсу користувача. Усі зазначені елементи управління в повному обсязі дають змогу керування комп'ютерною системою.

## 5. РЕЗУЛЬТАТИ РОБОТИ КОМП'ЮТЕРНОЇ СИСТЕМИ

В цьому розділі представлені результати роботи комп'ютерної системи. На першому рис 5.1 зображено роботу системи в режимі, інтегрованому в спеціальний автоматизований комплекс.

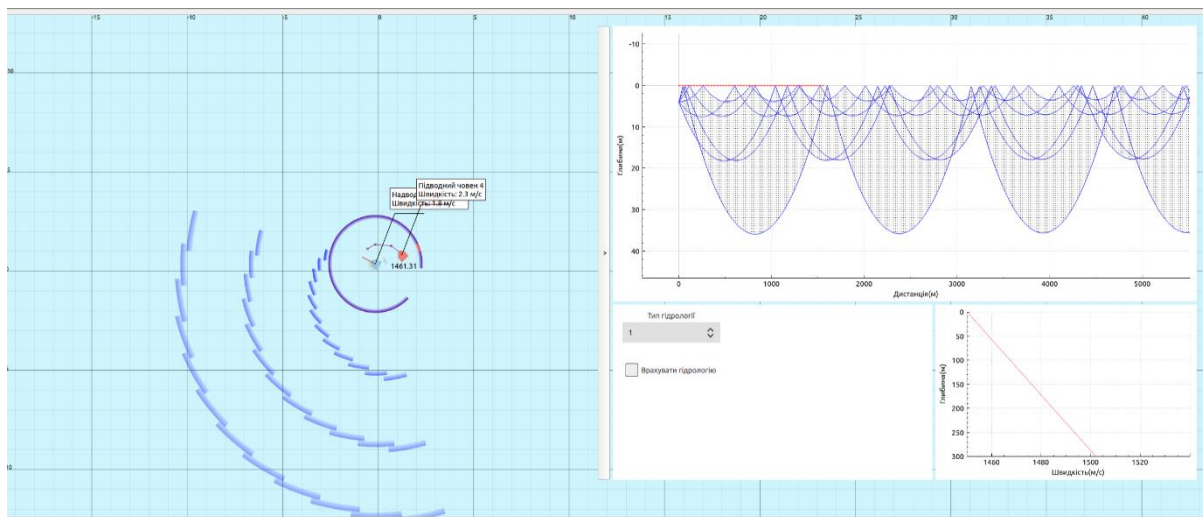


Рисунок 5.1 – Робота комп'ютерної системи в інтегрованому режимі

В залежності від режиму роботи інтерфейс користувача комп'ютерної системи змінюється. В інтегрованому режимі в інтерфейсі користувача залишається випадаюче меню зміни типу гідрології, в якому знаходиться лише перший, третій та шостий типи вертикального градієнту звуку, та кнопка вмикання чи вимикання роботи даної системи. Окрім цього, користувач має змогу наближувати або віддаляти графік результату. Графік результату перебудовується під час обирання конкретного підводного човна на сцені і дає змогу побачити результати для нього, а саме: чи є підводний човен в зоні освітленості, чи ні.

На рис 5.2 зображено роботу комп'ютерної системи окремо від спеціального комплексу. Як можна побачити, інтерфейс користувача майже не змінився, окрім того, що кількість елементів взаємодії збільшилась. Робота на одній глибині для різних типів ВРШЗ продемонстрована на рисунках 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 та 5.8.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

56

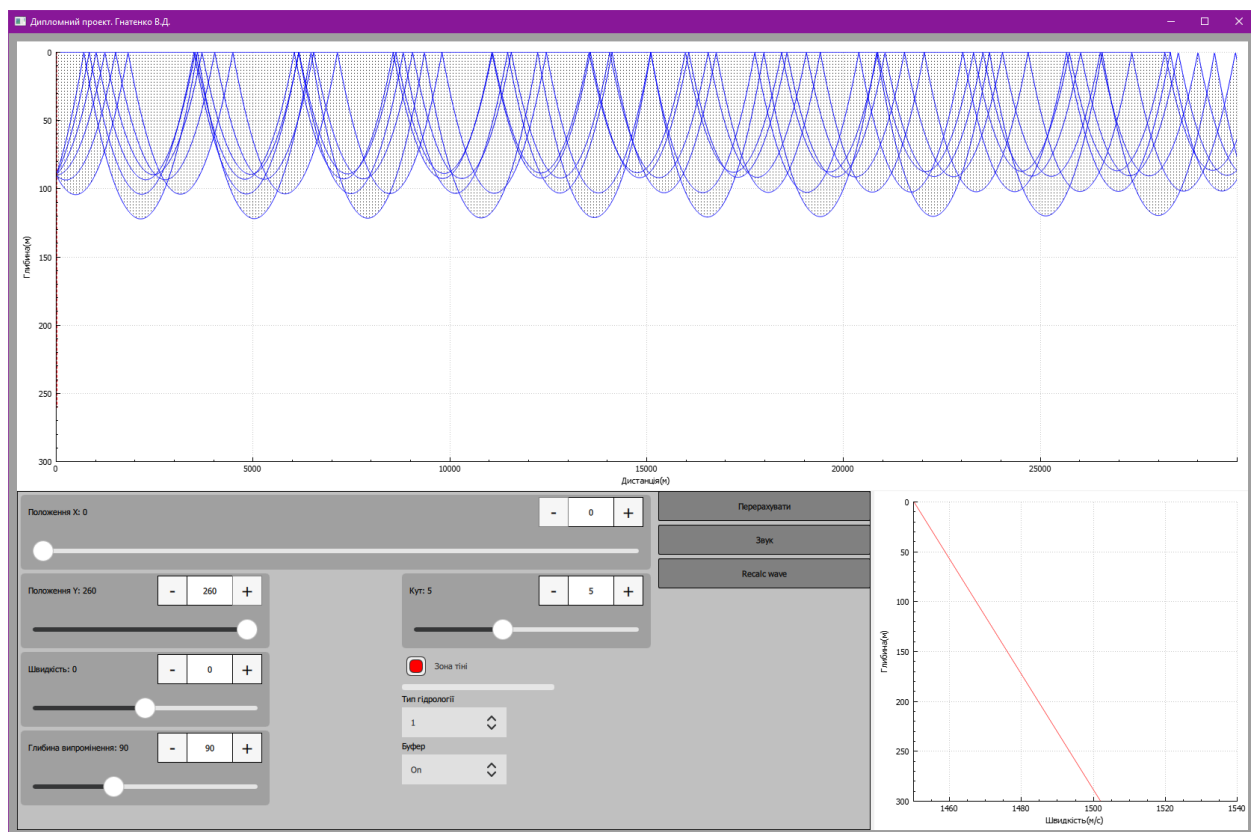


Рисунок 5.2 – Результат роботи при першому типі ВРШЗ

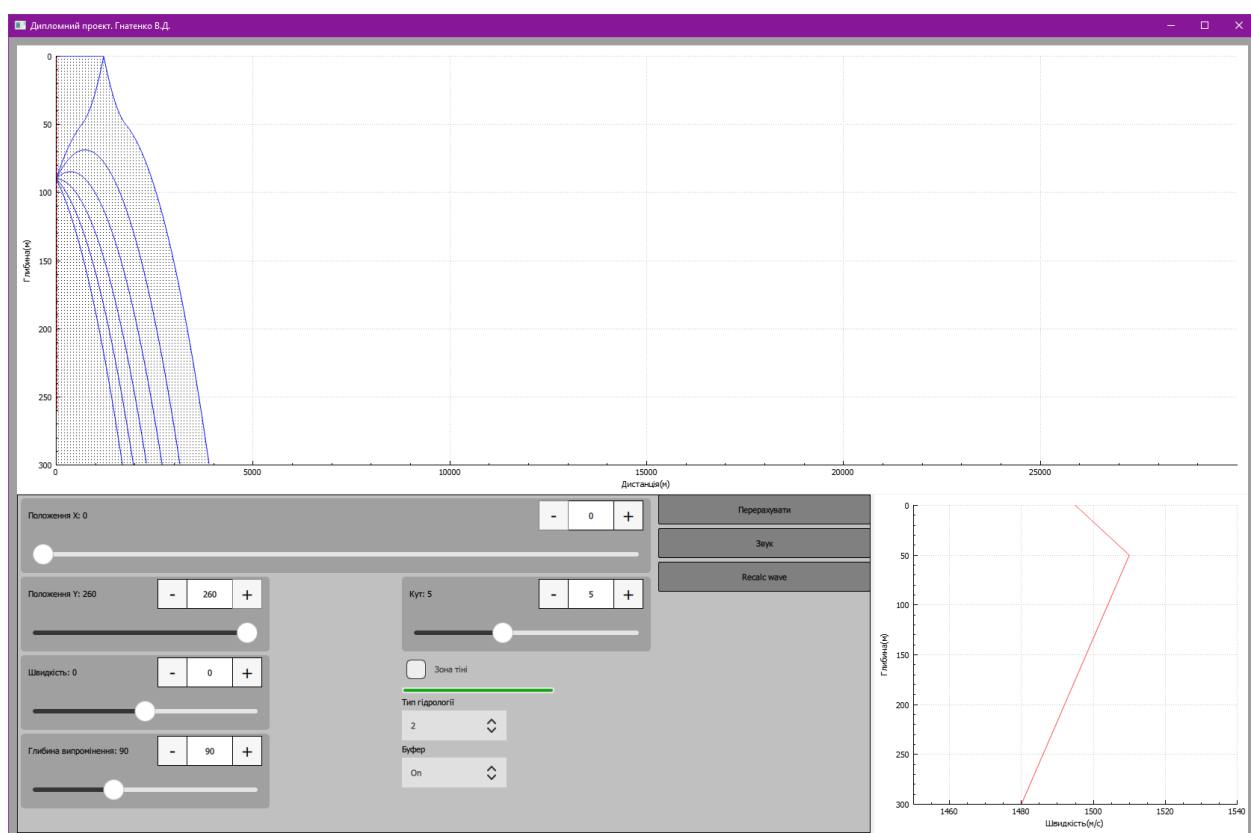


Рисунок 5.3 – Результат роботи при другому типі ВРШЗ

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

57

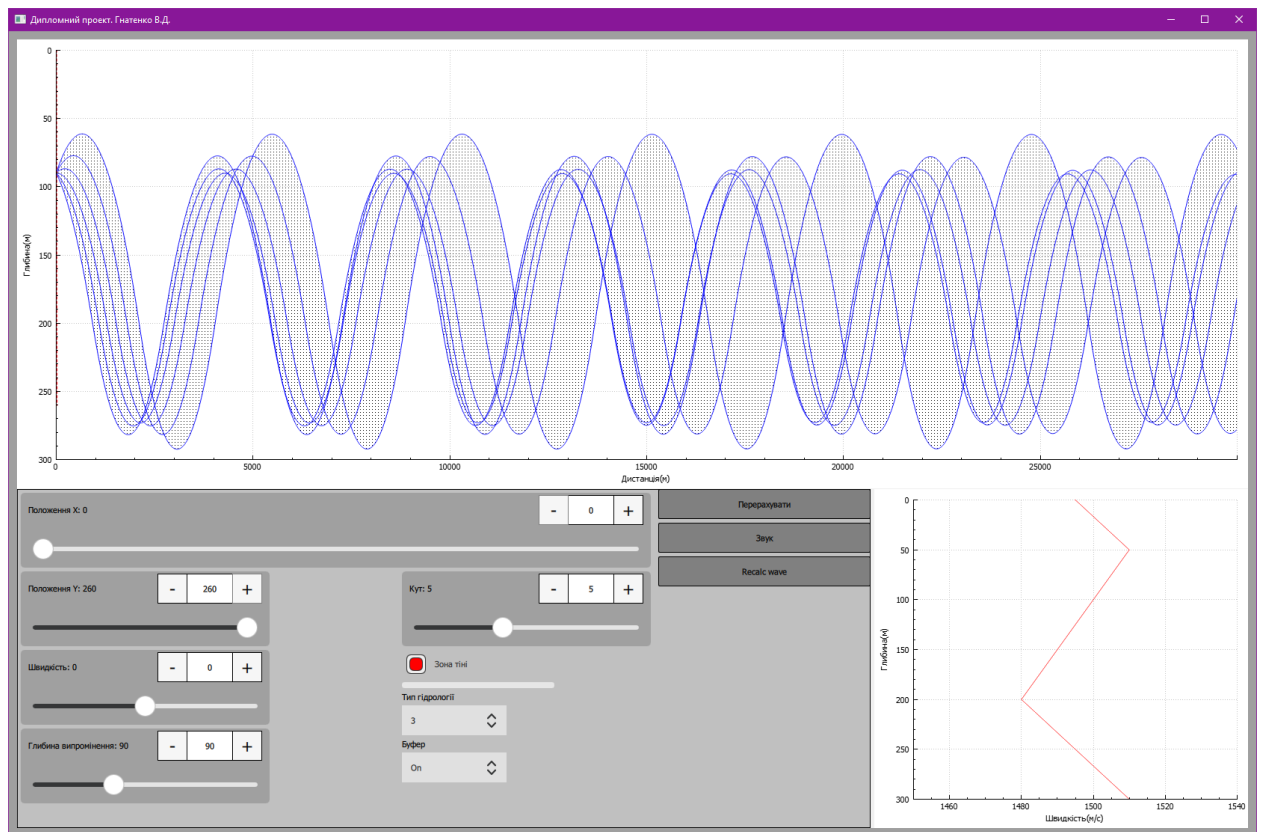


Рисунок 5.4 – Результат роботи при третьому типі ВРШЗ

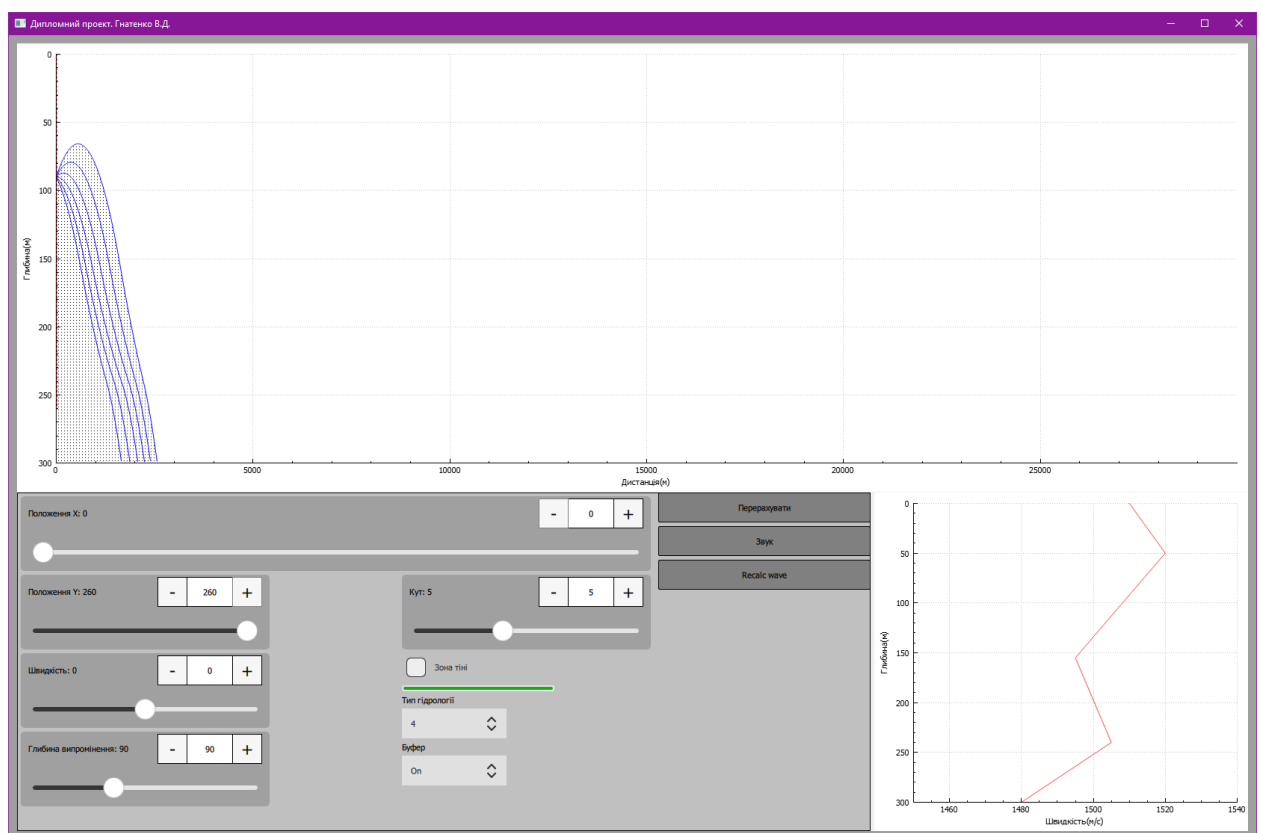


Рисунок 5.5 – Результат роботи при четвертому типі ВРШЗ

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

58

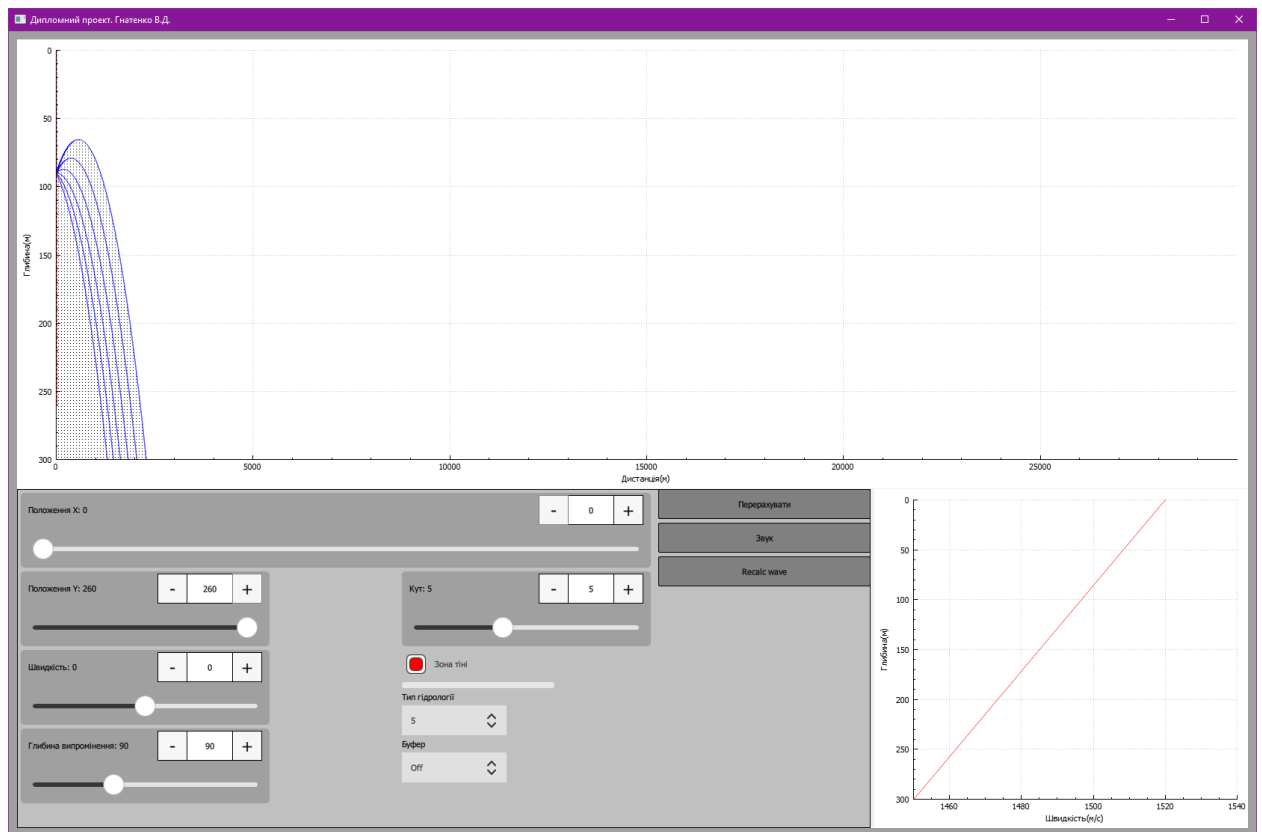


Рисунок 5.6 – Результат роботи при п'ятому типі ВРШЗ

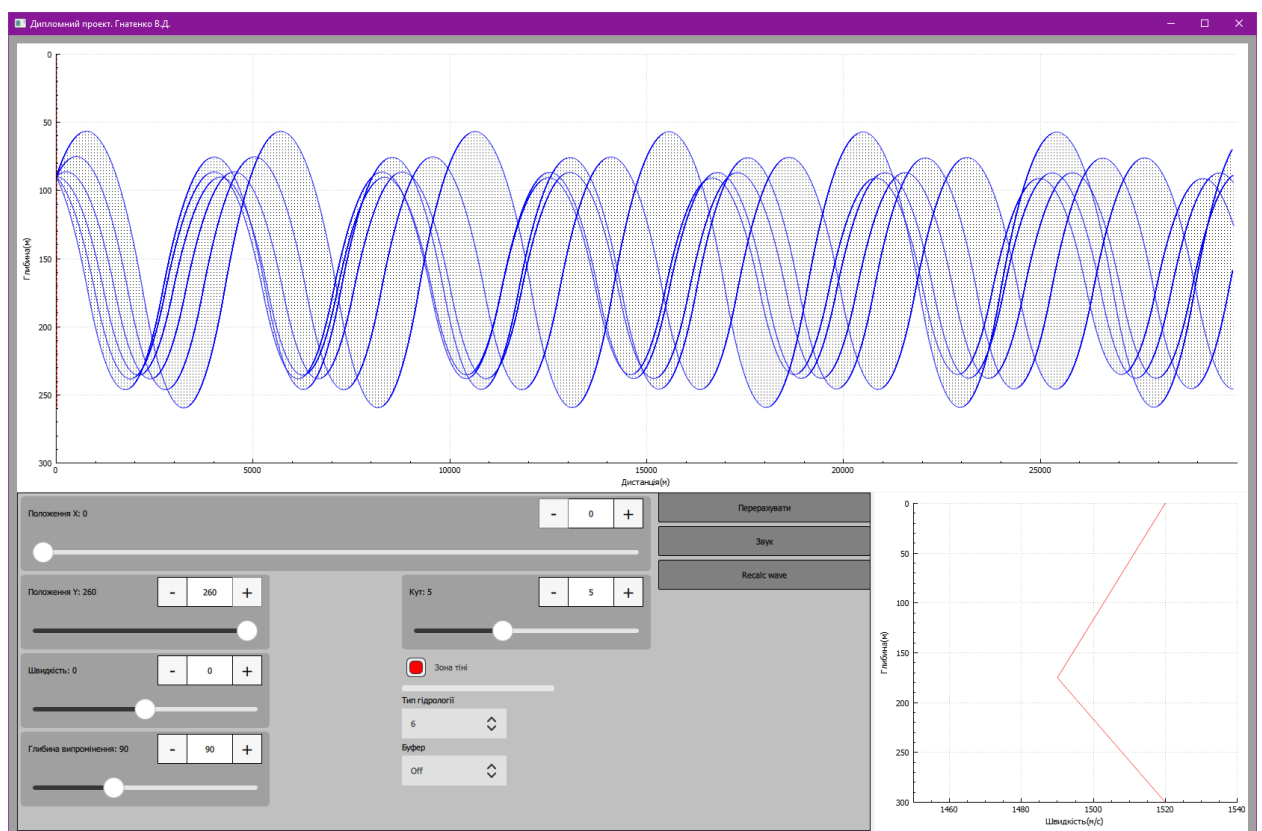


Рисунок 5.7 – Результат роботи при шостому типі ВРШЗ

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

59

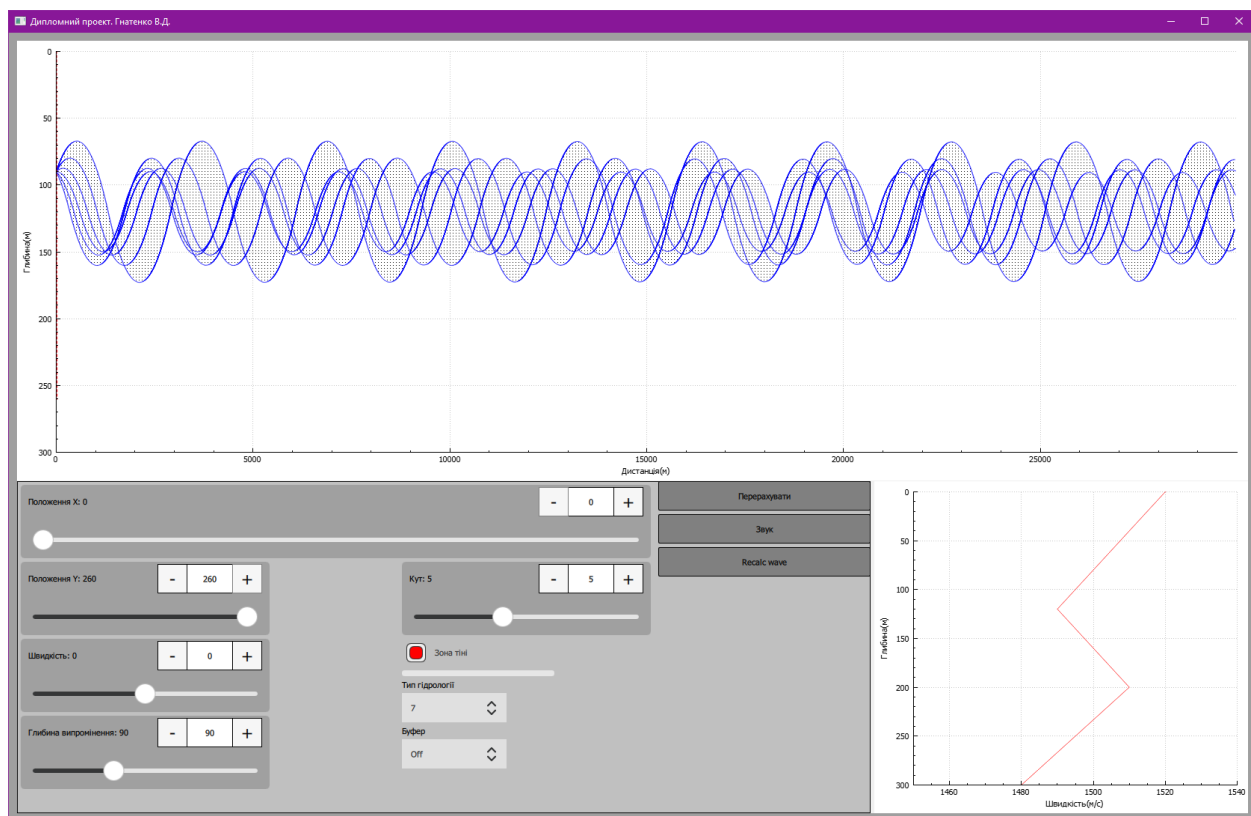


Рисунок 5.8 – Результат роботи при сьомому типі ВРШЗ

При попаданні підводного човна в зону освітленості, від його координат будується звукова хвиля, що йде у напрямку до станції (до  $X = 0$ ), що можна побачити на рис 5.9. Завдяки цьому можна побачити, яким чином і з якою траєкторією відбита хвиля (зображена червоним кольором) досягне приймача (початок випромінювання).

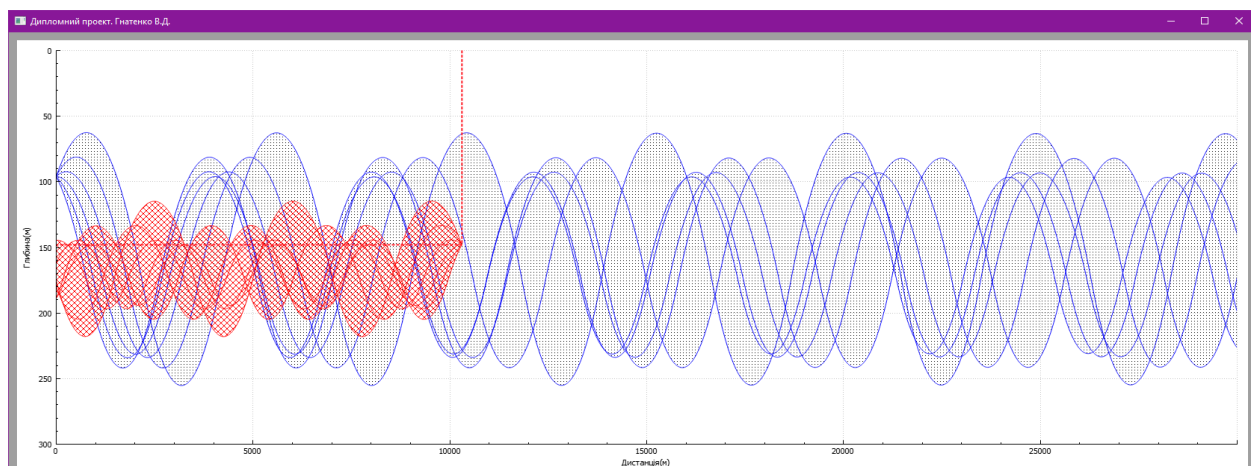


Рисунок 5.9 – Результат роботи при попаданні підводного човна в зону освітленості

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

60



На рис 5.9 видно, що точка випромінення не знаходиться в зоні освітленості. Це означає, що хоч звукова хвиля відбилась від цілі, але відбита хвиля не була прийнята, адже гідрофони, що знаходяться на місці антени, яка випромінює сигнал, не змогли отримати інформацію. Тому підводний човен не було знайдено.

На рис 5.10 зображено, як хвиля, від підводного човна відбивається та досягає гідрофонів на антені, що дає змогу знайти ціль.

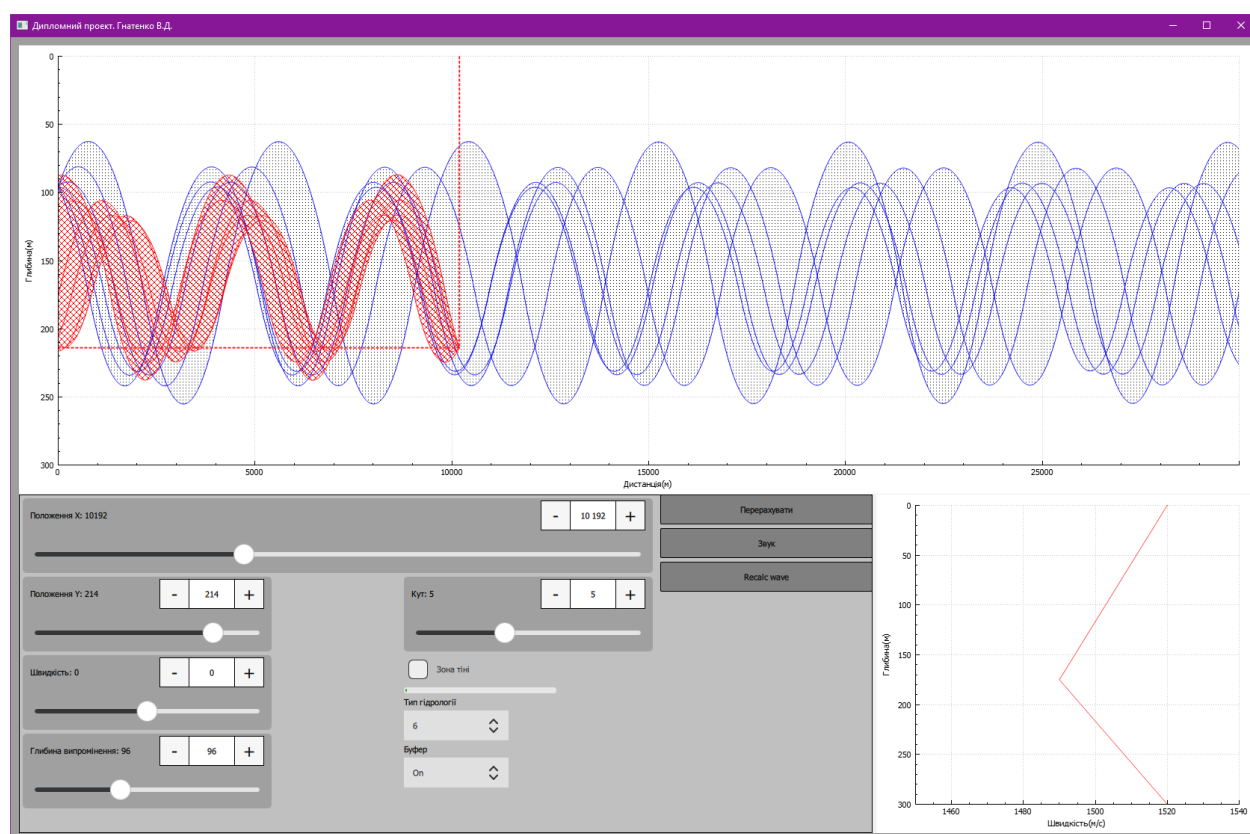


Рисунок 5.10 – Результат роботи при попаданні човна в зону освітленості

## Висновки до розділу

Спираючись на результати роботи комп'ютерної системи можна побачити, що імітація розповсюдження звукових хвиль відбувається таким самим чином, як і в реальному водному середовищі.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.002 ПЗ

Арк.

61



## ВИСНОВКИ

Головна мета дипломного проекту - створення комп'ютерної системи симуляції розповсюдження звукових хвиль в гідросфері. Розроблена комп'ютерна система може працювати як інтегрована система в складі спеціального автоматизованого комплексу для навчального класу підготовки операторів гідроакустичних засобів, так і в звичайному, незалежному режимі, що дозволяє набути та відпрацювати теоретичні навички майбутніх спеціалістів.

В результаті аналізу існуючих комп'ютерних систем було виявлено, що на території України не існує аналогів, які змогли б надати високу швидкодію та подібний функціонал. Під час проектування спеціального автоматизованого комплексу виникла потреба в імітації розповсюдження звукових хвиль в гідросфері для реалізації навчання в середовищі, близькому до реальних умов на кораблі.

На даний час, вищеописана комп'ютерна система використовується для навчання майбутніх спеціалістів акустичних систем в одному з навчальних центрів ВМС України, що дозволило значно зменшити витрати на забезпечення підготовки операторів гідроакустичних засобів ВМС України.

					ІАЛЦ.467200.002 ПЗ	Арк.
						62
Змін.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Finn B. Jensen William A. Kuperman, Michael B. Porter, Henrik Schmidt, Computational Ocean Acoustics. AIP Series in Modern Acoustics and Signal Processing, New York, 1994.
2. B. H. Porter M.B., Gaussian beam tracing for computing ocean acoustic fields, J. Acoust. Soc. America, 1987.
3. Qt - Електрон. дані (1 файл) – Режим доступу: <https://uk.wikipedia.org/wiki/Qt> (дата звернення 22.04.2020) – Назва з екрану.
4. A. D. Waite, Sonar for Practising Engineers, John Wiley & Sons Inc, 2002.
5. Richard P. Hodges, Underwater Acoustics: Analysis, Design and Performance of Sonar, 2011.
6. Ocean-acoustic propagation models - Електрон. дані (1 файл) – Режим доступу: <https://scripps.ucsd.edu/labs/buckingham/wp-content/uploads/sites/60/2015/04/1992Propagation.pdf> (дата звернення 23.04.2020) – Назва з екрану.
7. RKF45 - Електрон. дані (1 файл) – Режим доступу: [https://www.researchgate.net/publication/309645854\\_NUMERICAL\\_ACCURACY\\_BETWEEN\\_RUNGE-KUTTA\\_FEHLBERG\\_METHOD\\_AND\\_ADAMS-BASHFORTH\\_METHOD\\_FOR\\_FIRST\\_ORDER\\_ORDINARY\\_DIFFERENTIAL\\_EQUATIONS\\_WITH\\_BOUNDARY\\_VALUE\\_PROBL](https://www.researchgate.net/publication/309645854_NUMERICAL_ACCURACY_BETWEEN_RUNGE-KUTTA_FEHLBERG_METHOD_AND_ADAMS-BASHFORTH_METHOD_FOR_FIRST_ORDER_ORDINARY_DIFFERENTIAL_EQUATIONS_WITH_BOUNDARY_VALUE_PROBL) (дата звернення 23.04.2020) – Назва з екрану.
8. QCustomPlot - Електрон. дані (1 файл) – Режим доступу: <https://www.qcustomplot.com/> (дата звернення 23.04.2020) – Назва з екрану.
9. MUNC sound speed profile - Електрон. дані (1 файл) – Режим доступу: [https://www.researchgate.net/publication/234796174\\_Split\\_step\\_wavelet\\_G](https://www.researchgate.net/publication/234796174_Split_step_wavelet_G)

alarkin method based on parabolic equation model for solving underwa  
ter wave propag (дата звернення 23.04.2020) – Назва з екрану.

					ІАЛЦ.467200.002 ПЗ	Арк.
						64
Змін.	Арк.	№ докум.	Підпис	Дата		